



SEMANA 05

PROGRAMACIÓN DE APLICACIÓN

WEB Y MÓVIL

COLLACHAGUA NUÑEZ
Jessenia

ENUNCIADO 1

Sumatoria de numeros primos en un rango

Escribe un programa que solicite dos números y calcule la sumatoria de los números primos que existen entre dos valores. Utiliza un buen bucle for o while para recorrer los números en el rango y verifica si son primos.

Requerimientos

1. Entrada del usuario:

- El programa debe solicitar al usuario dos números enteros que definan el rango (el número inicial y el número final).
- Asegúrate de que el primer número sea menor o igual al segundo número.

2. Cálculo de números primos:

- Recorrer todos los números entre el primer número y el segundo número (inclusive).
- Verificar si cada número es primo.
- Los números primos son aquellos que solo tienen divisores 1 y el mismo número.

3. Sumatoria de números primos:

- Sumar todos los números primos encontrados dentro del rango.

4. Salida:

- Mostrar la suma de los números primos encontrados en el rango.

Pasos del programa:

1. Solicitar los dos números al usuario.
2. Usar un bucle for o while para recorrer todos los números en el rango.
3. Para cada número en el rango, verificar si es primo.
4. Sumar los números primos.
5. Mostrar el resultado final al usuario.

ENUNCIADO 2

Numeros de fibonacci hasta la N terminos

Implementa un programa que genere la secuencia de fibonacci hasta un numero n de términos ingresados por el usuario. Utiliza un bucle while o for para ir generando los números de la secuencia.

Requerimientos

1. Entrada del usuario:

- * El programa debe solicitar un número entero positivo n que indica cuántos términos de la secuencia de Fibonacci deben generarse.

2. Cálculo de la secuencia de Fibonacci:

- * La secuencia de Fibonacci comienza con 0 y 1, y cada número siguiente es la suma de los dos anteriores.

- * Ejemplo de la secuencia: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

3. Salida:

- * Mostrar la secuencia de Fibonacci con n términos en pantalla.

ENUNCIADO 3

Factorial de numeros grandes

Escribe un programa que calcule el factorial de un numero grande (por ejempl, 100) utilizando estructuras repetitivas y el tipo de datos BigInteger para manejar grandes números.

Requerimientos

1. Entrada del usuario:

- * El programa debe permitir que el usuario ingrese un número entero no negativo para calcular su factorial.

2. Cálculo del factorial:

- * Utilizar un bucle para multiplicar todos los números enteros desde 1 hasta el número ingresado.

3. Salida:

- * Mostrar el resultado del factorial.

ENUNCIADO 4

Inversión de un número

Crea un programa que invierta los dígitos de un número entero ingresado por el usuario, utilizando un bucle while para extraer y reordenar los dígitos.

Requerimientos

1. Entrada del usuario:

El programa debe permitir que el usuario ingrese un número entero.

2. Inversión de dígitos:

Utilizar un bucle while para extraer los dígitos del número y reordenarlos en el orden inverso.

3. Salida:

Mostrar el número invertido.

ENUNCIADO 5

Suma de matrices NxN

Escribe un programa que solicite dos matrices de tamaño $N \times N$ (donde N es proporcionado por el usuario) y luego realice la suma de las matrices utilizando bucles anidados for.

Requerimientos

1. Entrada del usuario:

Solicitar al usuario el tamaño NNN de las matrices.

Solicitar los elementos de ambas matrices.

2. Suma de matrices:

Utilizar bucles for para sumar las dos matrices.

3. Salida:

Mostrar la matriz resultante de la suma.

ENUNCIADO 6

Numero perfecto

Implementa un programa que encuentre y muestre todos los números perfectos entre 1 y 10000. Un número perfecto es aquel que es igual a la suma de sus divisores propios.

Requerimientos

1. Encontrar números perfectos:
Iterar desde 1 hasta 10,000.
Para cada número, calcular la suma de sus divisores propios.
- 2.-Salida:
Mostrar los números perfectos encontrados.

ENUNCIADO 7

Matriz de espiral

Crea un programa que imprima una matriz cuadrada de tamaño $n \times n$ en forma de espiral. Utiliza bucles anidados para recorrer las posiciones de la matriz en el orden adecuado.

Requerimientos

1. Entrada del usuario:
Solicitar al usuario el tamaño n de la matriz.
2. Llenar la matriz:
Llenar la matriz de forma espiral con números del 1 al n^2 .
3. Salida:
Mostrar la matriz resultante.

ENUNCIADO 8

Verificación de un número Armstrong

Escribe un programa que verifique si un número de n dígitos ingresado por el usuario es un número de Armstrong (o narcisista). Utiliza un bucle for para separar y elevar cada dígito a la potencia correspondiente.

Requerimientos

1. Encontrar números perfectos:
Iterar desde 1 hasta 10,000.
Para cada número, calcular la suma de sus divisores propios.
- 2.-Salida:
Mostrar los números perfectos encontrados.

ENUNCIADO 9

Cálculo de potencias usando multiplicación repetida

Crear un programa que calcule la potencia de un número usando multiplicación repetida, es decir, sin utilizar `pow`, y luego calcular la potencia utilizando un bucle while o for.

Requerimientos

1. Entrada del usuario:
2. Se solicita al usuario que ingrese la base y el exponente. Se utilizan `stdin.readLineSync()` y `int.parse()` para convertir la entrada a un número entero.
3. Inicialización de la potencia:
4. Se inicializa la variable potencia en 1, que será utilizada para almacenar el resultado del cálculo.
5. Cálculo de la potencia:
6. Se utiliza un bucle for que itera desde 0 hasta el valor del exponente. En cada iteración, se multiplica potencia por la base.
7. Salida:
8. Se imprime el resultado final, mostrando la base, el exponente y la potencia calculada.