Hoja 3. Programación (ejercicios avanzados usando estructuras de control)

Ejercicios sobre tipos de números

1. Realiza un método que, dado un número de tres cifras, averigüe si es un **número Armstrong o Narcisista.** Un número es Armstrong cuando la suma de cada uno de los números que lo componen elevado al número de dígitos de dicho número de dicho número da como resultado el propio número. Ejemplo de un número de 3 dígitos: $153=1^3+5^3+3^3$.

2. **Número Perfecto**. Escribe un método en Java que determine si un número es un "número perfecto". Un número es perfecto si la suma de sus divisores propios (excluyendo a sí mismo) es igual al número. Por ejemplo, 28 es un número perfecto porque sus divisores propios son 1, 2, 4, 7, y la suma de ellos es 1 + 2 + 4 + 7 = 14, que es igual a 28.

3. **Números Primos Gemelos**. Desarrolla un programa que encuentre y muestre todos los pares de "números primos gemelos" en un rango dado. Los números primos gemelos son dos números primos que difieren en 2 unidades. Por ejemplo, 41 y 43 son un par de números primos gemelos.

4. **Suma de Números Primos**. Crea un programa en Java que encuentre todas las formas en que un número entero positivo puede expresarse como la suma de dos números primos. Por ejemplo, 10 se puede expresar como 3 + 7 y como 5 + 5.

5. **Secuencia de Fibonacci.** Desarrolla un programa que genere los primeros n términos de la secuencia de Fibonacci. La secuencia de Fibonacci comienza con 0 y 1, y los términos subsiguientes son la suma de los dos términos anteriores (0, 1, 1, 2, 3, 5, 8, ...).

6**. Cálculo de Divisores Primos**. Escribe un programa en Java que, dada un número entero positivo, calcule y muestre en pantalla todos los divisores primos de ese número. Un divisor primo es aquel que es un número primo y divide exactamente al número original. El programa debe calcular si un número es primo y luego encontrar los divisores primos del número dado.
Requisitos:

- El programa debe solicitar al usuario un número entero positivo.
- Debe verificar si el número ingresado es válido (mayor que 0).
- Debe calcular y mostrar en pantalla todos los divisores primos del número ingresado.

Enunciado 7: Factorial de un Número

7. Escribe un método que **calcule el factorial** de un número entero no negativo. El factorial de un número n se define como la multiplicación de todos los números enteros positivos desde 1 hasta n.

8. **Cálculo de Números Amigos.** Desarrolla un programa en Java que encuentre y muestre pares de "números amigos". Dos números enteros positivos A y B se consideran amigos si la suma de los divisores propios de A (excluyendo a sí mismo) es igual a B, y viceversa.
Por ejemplo, los números 220 y 284 son un par de números amigos. La suma de los divisores propios de 220 (1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110) es igual a 284, y la suma de los divisores propios de 284 (1 + 2 + 4 + 71 + 142) es igual a 220.
Requisitos:

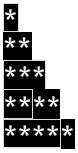
- El programa debe permitir al usuario especificar un rango de números enteros positivos para buscar números amigos.

- Debe encontrar y mostrar todos los pares de números amigos dentro del rango especificado por el usuario.
- El programa debe ser eficiente y manejar el cálculo de divisores propios de manera adecuada.

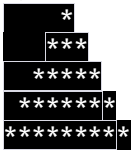
Ejercicios sobre figuras (búsqueda de patrones)

1.- Realiza un programa usando bucles que muestren las siguientes figuras. Se deben pedir por teclado el número de filas de la figura antes de dibujarlas.

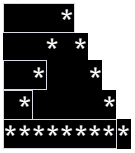
a) Triángulo de asteriscos.



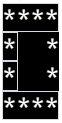
b) Pirámide de asteriscos.



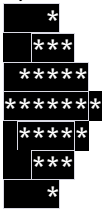
c) Patrón de pirámide hueca



d) Patrón de cuadrado hueco



e) Patrón de diamante



f) Patrón de diamante hueco