Actividad 4A: Ejercicios

(actividad para realizar en clase/extraclase)

Solucionar cada uno de los problemas planteados usando un lenguaje de programación y evaluando si es posible reutilizar código llamando funciones ya implementadas en los problemas anteriores de esta misma actividad.

01. Implemente una función llamada **esMultiplo** que reciba como parámetros dos números enteros y retorne si el primero es un múltiplo del segundo.

Implemente una función llamada **esDivisor** que reciba como parámetros dos números enteros y retorne si el primero es un divisor del segundo.

02. Implemente una función llamada **esPar** que reciba como parámetro un número entero y retorne si este es o no un número par.

03. Implemente una función llamada **esImpar** que reciba como parámetro un número entero y retorne si este es o no un número impar.

04. Implemente una función llamada **esPositivo** que reciba como parámetro un número entero y retorne si este es o no un número positivo.

05. Implemente una función llamada **esNegativo** que reciba como parámetro un número entero y retorne si este es o no un número negativo.

06. Implemente una función llamada **mayor** que reciba como parámetros dos números enteros y retorne el mayor de estos.

07. Implemente una función llamada **menor** que reciba como parámetro dos números enteros y retorne el menor de estos.

08. Implemente una función llamada **factorial** que reciba como parámetros un número entero y retorne el factorial de este.

09. Implemente una función llamada **dobleFactorial** que reciba como parámetros un número entero y retorne el doble factorial de este.

10. Implemente una función llamada **cantDivisoresPos** que reciba como parámetro un número entero y retorne la cantidad de divisores postitivos que tiene ese número.

11. Implemente una función llamada **cantDivisoresNeg** que reciba como parámetro un número entero y retorne la cantidad de divisores negativos que tiene ese número.

12. Implemente una función llamada **cantDivisores** que reciba como parámetro un número entero y retorne la cantidad de divisores que tiene ese número.

13. Implemente una función llamada **sumDivisoresPos** que reciba como parámetro un número entero y retorne la suma de sus divisores positivos.

14. Implemente una función llamada **sumDivisoresNeg** que reciba como parámetro un número entero y retorne la suma de sus divisores negativos.

15. Implemente una función llamada **sumDivisores** que reciba como parámetro un número entero y retorne la suma de sus divisores.

16. Implemente una función llamada **esCompuesto** que reciba como parámetros un número entero y retorne si este es o no es un número compuesto.

17. Implemente una función llamada **esPrimo** que reciba como parámetros un número entero y retorne si este es o no es un número primo.

18. Implemente una función llamada **primorial** que reciba como parámetros un número entero y retorne el primorial de este.

19. Implemente una función llamada **esPerfecto** que reciba como parámetro un número entero y diga si este es o no es un número perfecto.

20. Implemente una función llamada **sonAmigos** que reciba como parámetros dos números enteros y retorne si estos son o no números amigos.