**Números Perfectos**

Programación II

**Números Perfectos**

Un número perfecto es un entero positivo que es equivalente a la suma de todos sus divisores excluyendo al número en si. Por ejemplo, el número 6 es perfecto debido a que sus divisores son 1,2,3 (excluyendo al número) y la suma de ellos es 1+2+3=6.

**Ejercicio**

Diseñe un programa que solicite un número al usuario e indique si dicho numero es perfecto o no.

**Explorando Ideas**

1. Modifique su programa para que ahora encuentre los números perfectos menores al numero ingresado. Por ejemplo, si el usuario ingresa el numero 100, entonces el programa debe mostrar en pantalla que ha encontrado dos números perfectos: 6 y 28.
2. ¿Qué sucede cuando el numero ingresado es muy grande?, por ejemplo 1,000,000

Tarda demasiado

1. ¿Cuál es el orden del algoritmo implementado? (Es más, ¿Qué es el orden de un algoritmo?)

O(n2). El orden de complejidad de un algoritmo mide cuán rápido aumenta el tiempo de ejecución del algoritmo cuando aumentan los datos de entrada.

1. ¿Qué características del programa generan los comportamientos experimentados?

Que la complejidad de mi función *isPerfecto* es O(n) y se manda a llamar la función O(n) veces.

1. Actualmente se están destinando esfuerzos para el desarrollo de Quantum Computing, si implementáramos una solución como un algoritmo en Quantum Computing, ¿Podría mejorar la ejecución del programa?

Si mejora no creo que sea una mejora significante porque mi algoritmo no factoriza números, que es donde se ha visto una mejora exponencial en la computación cuántica respecto a la computación convencional