

IWI-131 Programación – Tarea UVA 5: Funciones

Números Especiales. En el universo de los enteros positivos existen algunos números que presentan ciertas características especiales. Para esta tarea consideraremos los números primos, los primos de Mersenne, los perfectos, los narcisistas y los amigos.

1. **Número primo:** es aquel divisible solo por 1 y por el mismo número. Por ejemplo 7 es primo ya que solo es divisible por 1 y por 7. En cambio 8 no es primo, ya que pese a ser divisible por 1 y por 8, también lo es por 2 y por 4.
2. **Número primo de Mersenne:** es un número primo que puede ser representado como $2^n - 1$, siendo n algún número entero positivo. Por ejemplo, 31 es un número primo de Mersenne ya que es primo y puede escribirse como $2^5 - 1$, siendo $n = 5$. Por otro lado, 13 no es un primo de Mersenne ya que, pese a ser primo, no existe un n entero positivo para el cual se cumple $2^n - 1 = 13$. Para más detalle sobre estos números puede ver el siguiente video: <https://youtu.be/T0xKHwQH-4I?t=128>
3. **Número perfecto:** es aquel que es igual a la suma de sus divisores propios. Entiéndase por divisores propios de un número entero positivo mayor que uno a todos los enteros que lo dividen de forma exacta (resto de la división es cero) sin considerar al mismo número. Por ejemplo, los divisores propios de 6 son 1, 2 y 3. Los de 28 son 1, 2, 4, 7 y 14. Justamente 6 es un número perfecto ya que es igual a $1 + 2 + 3$, la suma de sus divisores propios. Lo mismo ocurre con 28 ya que $1 + 2 + 4 + 7 + 14 = 28$, por lo tanto 28 también es perfecto. No ocurre lo mismo con 32, ya que la suma de sus divisores propios es $1 + 2 + 4 + 8 + 16 = 31$. Más sobre estos números en el siguiente video: <https://www.youtube.com/watch?v=T0xKHwQH-4I>
4. **Número narcisista:** es aquel que es igual a la suma de sus dígitos, elevando cada uno a la cantidad de cifras que tiene. Por ejemplo 153 es un número narcisista ya que es igual a $1^3 + 5^3 + 3^3$. Lo mismo ocurre con 1634, ya que es igual a $1^4 + 6^4 + 3^4 + 4^4$. Más sobre estos números en: <https://www.youtube.com/watch?v=4aMtJ-V26Z4>
5. **Números amigos:** dos números enteros, digamos n y m , son amigos entre sí si la suma de los divisores propios de n es igual a m y la suma de los divisores propios de m es igual a n . Por ejemplo, 220 y 284 son amigos ya que la suma de los divisores propios de 220 es $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$ mientras que la suma de los divisores propios de 284 es $1 + 2 + 4 + 71 + 142 = 220$. Más sobre estos números en: <https://www.youtube.com/watch?v=fUSZBVYZdKY>

Un grupo de profesores ha recibido un cuantioso contrato para desarrollar una aplicación en Python que solicite al usuario dos números enteros y determine si estos son especiales en alguna de las categorías ya mencionadas.

Un uso importante de las funciones en la industria de desarrollo de software es el poder separar un problema complejo en partes más simples y manejables. Este grupo de profesores ha desarrollado la entrada y salida de datos (los `input()` y los `print()`) y ha puesto a disposición de usted este código en un archivo llamado `principal.py`. Desde este archivo se llaman diversas funciones para resolver el problema al que nos enfrentamos. Este archivo no debe ser modificado y por lo tanto tampoco debe ser entregado con la solución.

A su vez también pone a su disposición un archivo llamado `funciones.py`, que incluye las siguientes funciones que son utilizadas desde `principal.py`:

1. `primo(n)`: función que recibe un número entero `n` y retorna **True** si es primo y **False** en caso contrario; esta función ya se encuentra desarrollada y probada en su totalidad.
2. `primo_de_mersenne(n)`: función que recibe un número entero `n` y retorna **True** si es un número primo de mersenne y **False** en caso contrario. Usted debe desarrollar esta función.
3. `numero_perfecto(n)`: función que recibe un número entero `n` y retorna **True** si es un número perfecto y **False** en caso contrario. Usted debe desarrollar esta función.
4. `numero_narcisista(n)`: función que recibe un número entero `n` y retorna **True** si es un número narcisista y **False** en caso contrario. Usted debe desarrollar esta función.
5. `son_amigos(n,m)`: función que recibe dos números enteros `n` y `m` que retorna **True** si ambos números son amigos y **False** en caso contrario. Usted debe desarrollar esta función.

Como bono, se incluye también la implementación de la función `suma_divisores_propios` que recibe un número entero `n` y retorna la suma de sus divisores propios. Usted puede decidir si esta función le es de utilidad o no.

Entrega: Solo debe modificar el archivo `funciones.py`. En este debe definir cada función solicitada. Recuerde que no debe usar ni `input()` ni `print()`, pues esto ya se encuentra implementado en el archivo `principal.py`. Puede incluir tantas funciones adicionales en el archivo `funciones.py` como sea necesario, pero deben estar como mínimo las 5 funciones solicitadas.

Sobre ejecución del programa: Se debe ejecutar el archivo `principal.py`. Éste debe estar en la misma carpeta que el archivo `funciones.py`. Si ambos están en la misma carpeta, al ejecutar `principal.py` se le solicitará al usuario ingresar dos números enteros mayores que uno. Puede verificar esto apenas descargue los archivos de la tarea. Como las funciones no están implementadas todavía, el programa inicialmente solo pide datos y nada más.

Un ejemplo de ejecución con el programa base, sin haber hecho nada, sería:

```
Ingrese un numero entero mayor que uno: 13
Ingresa otro numero entero mayor que uno: 31
```

En cambio, este mismo ejemplo luego de implementar las 5 funciones solicitadas en el archivo `funciones.py` debe mostrar:

```
Ingrese un numero entero mayor que uno: 13
13 es primo
Ingresa otro numero entero mayor que uno: 31
31 es primo
31 es primo de mersenne
```

Recuerde, una vez mas, que usted no debe incluir ningún `input()` ni ningún `print()` ya que toda la interacción con el usuario ya se encuentra implementada en el archivo `principal.py`, el cual no debe modificar.