Práctica 4.1

Implementa una función **ullman** en PHP en un fichero **ullman.inc.php** en el que reciba un entero mayor que 1, que introducirá el usuario por teclado desde el programa principal (volveremos a pedir el número en caso de no ser mayor de 1 o no ser un número), calcule e **imprima por pantalla**los elementos correspondientes a la **Conjetura de Ullman**.

La conjetura consiste en lo siguiente:

·         Empieza con cualquier entero positivo mayor que 1.

·         Si es par, se divide entre 2.

·         Si es impar se multiplica por 3 y se le suma 1.

·         Se repite hasta obtener el número 1.

Al final se obtendrá el número 1, independientemente del entero inicial.

Por ejemplo, cuando el entero inicial es 26, la secuencia será: 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

## Práctica 4.2

Crear un fichero en PHP que incluya una **función**con un **rellenador**de boletos de la primitiva.

Cada apuesta consistirá en 6 números entre 1 y 49 en la que habremos de evitar la posibilidad de que un número se repita dos veces.

Una vez conseguido esto repetir este proceso para un boleto entero (para las 8 apuestas en total de las que consta un boleto)

## Práctica 4.3

Un número **abundante** es un número natural (entero mayor que 0) que es **menor** que la **suma** de sus **divisores** (excepto el propio número).

* Ej: 12 -> sus divisores son 1, 2, 3, 4, 6 -> 1+2+3+4+6=16>12

Así, los primeros números abundantes son: 12, 18, 24 y 30.

Hacer una **función** **abundante** en un fichero PHP que **devuelva** **true** si un número pasado como argumento es abundante **o** **false** en caso contrario.

Práctica 4.4

Un **número** **defectivo** o deficiente es un **número natural** que es **mayor** que la **suma** de sus **divisores** **propios** exceptuándose a sí mismo.

* Ejemplo: **7 es defectivo**, ya que los divisores de 7 (exceptuando el 7) son 1 à por tanto el número es mayor que la suma de sus divisores, por lo que es defectivo.
* Ejemplo **4 es defectivo**, ya que sus divisores son 1,2 à 1+2=3 à 4 >3 à por tanto, es defectivo
* Ejemplo **20 no es defectivo**, ya que sus divisores son: 1,2,4,5,10 à 1+2+4+5+10=22 à 20 <22 à no defectivo

 Hacer una **función** que devuelva **true** o **false** en caso de que un **número** pasado por **parámetro** **sea** **defectivo** o no.

**Mostar por pantalla** del **1 al 10.000** solo **aquellos** **números** que son **defectivos** o deficientes. Solo mostrará los defectivos, los que no lo son no deben mostrarse.