Práctica 4.1 (Ejercicio 1)

Implementa una función JavaScript 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 (Ejercicio 2)

El 11 del 11 es un día especial, el día del sorteo único de la ONCE. Este sorteo extraordinario tiene un espectacular reparto de premios:

* un primer premio de 11.000.000 €,
* 11 premios de 1.000.000 €

 El **cupón ganador** del premio principal estará formado por un **número de cinco cifras** entre el 00000 y el 99999 y un número de serie comprendido entre el 001 y el 120.

Además de este habrá **11 premios de 1.000.000** **€** . Habrá que comprobar que ninguno de estos 11 premios ninguno coincida con el cupón ganador (solo con el ganador, con el resto no es necesario)

Este año con la crisis del COVID-19 en lugar de 5 bombos con los números del 0 al 9, nos han pedido realizar una aplicación que genere de forma aleatoria dichos números. Habrá un sexto bombo en el que aparecerá la serie (desde 1 hasta 120).

Hacer **un programa que genere los cinco números del premio gordo** (**uno a uno**) **y la serie**, y los **11 premios de 1 millón de euros (se pueden generar completos)**, que serán mostrados por pantalla.

**Debe quedar bien claro cuál es el premio gordo (y su serie) y cuáles son los 11 premios de 1 millón de euros (con su serie).**

## Práctica 4.3 (Ejercicio 3)

Crear un programa en JavaScript que sea 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 4.4 (Ejercicio 4)

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** que **devuelva** **true** si un número pasado como argumento es abundante **o** **false** en caso contrario.

## Práctica 4.5 (Ejercicio 1)

Hacer un programa en que funcione como una máquina tragaperras antigua.

Generaremos 3 números aleatorios del 1 al 10.

Implementar una **función** **comprobarPremio,**que recibe los tres números generados y se comportará de la siguiente manera:

* Si **2 de los 3 números son iguales habremos ganado 100** **€** (mostrará ese mensaje por pantalla con un **alert**).
* **Si los 3 resultados son iguales habremos ganado 500 €** (mostrará ese mensaje por pantalla mediante un **alert**)
* Si los **3 números son distintos** mostrará el mensaje mediante **alert**: “**Siga buscando, pardillo**”

## Práctica 4.6 (Ejercicio 2)

Vamos a desarrollar la sucesión de Juan Ferrer, en honor a su creador, YO:

* Los cinco primeros términos de la sucesión son 1
* El sexto término se calcula sumando el término 1, 2 y 3 menos el término 4.
* El séptimo término será la suma del 2, 3 y 4 elemento menos el 5, y así sucesivamente.

 Los primeros términos de esta sucesión son: 1,1,1,1,1,2,2,1,2,4,3,1,4, …

Mostrar por pantalla los primeros 1000 términos de la sucesión de Juan.

## Práctica 4.7 (Ejercicio 3)

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.

## Práctica 4.8 (Ejercicio 4)

Este año para el sorteo de Navidad de la lotería, los niños de San Ildefonso se han infectado todos de COVID-19, y por tanto el sorteo corre el riesgo de ser suspendido.

Como somos el primo listo del organizador del sorteo nos ha pedido ayuda para la realización de este y por tanto vamos a desarrollar un **programa que nos permita generar los premios de dicho sorteo**.

Los citados premios son los siguientes:

* **1 primer premio**, llamado “el Gordo”, de 4 millones de euros
* **1 segundo premio**, de 1.250.000 € (hay que comprobar que no coincide con el gordo)
* **1 tercer premio** de 500.00 € (hay que comprobar que no coincide con el gordo)
* **2 cuartos premios** con 200.000 € cada uno (hay que comprobar que no coincide con el gordo)
* **8 quintos premios** con 60.000 € cada uno (hay que comprobar que no coincide con el gordo)
* 1.794 premios de 1000 €

 Como me he quejado a mi primo ya que me ha avisado el día de antes, me indica que para considerar el ejercicio como correcto, **no será obligatorio generar los 1.794 premios de la llamada “pedrea**”, pero que si los generamos seremos recompensados con un punto extra sobre la nota que obtengamos (siempre y cuando la suma de la nota más el punto no sea superior a 10, en cuyo caso tendremos un 10).