

EJERCICIOS DE TÉCNICAS BÁSICAS PROGRAMACIÓN, funciones

Nota: es todos estos ejercicios la interfaz debe ser lo más sencilla posible, en el sentido de que lo que importa es que el *script* introduzca las entradas indicadas y las salidas, pero sin preocuparse de detalles de presentación.

1. Desarrolla un script que tenga una función que permite comprobar si un número es primo o no. Entonces el script, utilizando dicha función, creará una tabla de los números desde el 1 hasta el 100, formateada en 10 filas de 10 números cada una, y si un número es primo lo imprimirá y si no lo es, imprimirá --.

Por ejemplo:

```
1  2  3  --  5  --  7  --  --  --  
11 -- 13 -- -- -- 17 -- 19 --  
etcétera hasta -- (100)
```

2. Utilizando la función `rand()` crea un script que genere una tabla con un número aleatorio de filas comprendido entre 1 y 100.

- ⑩ en cada fila los números se generarán de forma aleatoria
- ⑩ habrá tantos números por fila como el valor que tenga el primer número aleatorio de la fila; por ejemplo:

```
7 45 23 43 12 12 1
```

- ⑩ los números a mostrar, también *random*, estarán comprendidos entre 1 y 50.

3. Desarrolla un script que tenga una función que permite resolver una ecuación de segundo grado $ax^2 + bx + c = 0$. La función recibirá como parámetros los 3 coeficientes a, b y c y devuelva como resultado un objeto con las 2 soluciones raíces o con NaN como soluciones. Una ecuación de segundo grado no tiene soluciones reales si su discriminante es negativo: $b^2 - 4ac < 0$.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Después añade un bucle que genere al azar 10 tripletas (a, b, c) y muestre para cada tripleta si tiene solución y cuál, o si no tiene soluciones reales. Una tripleta y su solución por línea.

4. La serie de Fibonacci es una serie de números enteros positivos: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... que comienza por los términos 0 y 1 y tiene la propiedad de que cada término posterior es la suma de sus 2 predecesores en la sucesión ($2 = 1 + 1$, $3 = 2 + 1$, ...).

Escribe 2 funciones `fibonacci1` y `fibonacci2` que calculen el término n de la sucesión que reciben como parámetro:

- ⑩ `fibonacci1`: lo hará de forma iterativa
- ⑩ `fibonacci2`: lo hará de forma recursiva

Prueba cada una de las funciones con los números siguientes: 10, 15, 20, 25.

5. Desarrolla una función que recibe un número entero positivo y lo devuelve como una cadena de texto formada por sus dígitos separados por 4 espacios. Por ejemplo: 455334 ---> "4 5 5 3 4 4"

Comprueba la función generando 10 números al azar entre 9999 y 1000000 e invocando para cada uno la función. Cada resultado debe aparecer en una nueva línea.