

<b>Nombre y apellidos:</b>		<b>Fecha:</b>	<b>27/11/2019</b>
<b>Módulo</b>	Programación (examen 1ª evaluación)	<b>CFGS</b>	<b>DAW</b>

### **Normativa:**

- El examen se debe resolver con los conceptos vistos en clase: funciones, procedimientos, if, elif, else, for, while, listas, diccionarios, cadenas, tuplas...
- Se pueden realizar más funciones de las indicadas.
- No se puede abrir el navegador, así como cualquier tipo de software de comunicación. El incumplimiento de esta norma es motivo de finalización del examen, y por tanto, no será corregido.
- Se pueden consultar los pdf del aula virtual mediante un lector de pdf (excluido el navegador), así como los programas realizados durante las prácticas.
- Nada más iniciar el examen lo primero que debéis incluir es una cabecera que aparezca: #nombre apellido apellido2 – examen 1º evaluación.
- Se creará un fichero .py por cada ejercicio con el siguiente nombre ej1nombreapellido.py ej2nombreapellido.py
- Al finalizar, se levantará la mano y se entregará los ficheros .py en la entrega programada en Classroom.

**Criterio de evaluación:** Cada apartado en el que aparezca la puntuación se evaluará de la siguiente manera:

- 80% de la puntuación si funciona.
- 10% de la puntuación si se usan las estructuras de datos adecuadas y éstas se utilizan adecuadamente.
- 10% de la puntuación calidad del código: nombrado de variables, estructuras de datos y funciones; comentarios en el código, etc.

### **1. La conjetura de Sheldon Cooper (5 puntos)**

Un estudio reciente ha demostrado que efectivamente, el número 73 tiene unas características concretas que ningún otro número tiene. En concreto, Sheldon Cooper en la serie Big Bang Theory dijo: *“El 73 es el 21º número primo, y leído al revés es el 37, que es el 12º número primo, que al revés es el 21, que es el resultado de multiplicar, agarraos fuerte, 7 por 3.”* Apoyándote en esto, se pide un programa que valide si dado un número, éste es un número que cumple la conjetura de Sheldon, mostrando dicho mensaje por pantalla.

Teniendo en cuenta esto, desarrolla las siguiente funciones/procedimientos que te servirán de apoyo para resolver el problema:

**Función esPrimo** (0,5 puntos). Esta función recibirá como parámetro un número entero, y devolverá un booleano indicando si es primo (cierto) o no lo es (falso).

Función **obtenerEspejo** (1 punto). Esta función recibe como parámetro un número y devuelve ese número dado la vuelta, es decir el espejo. Por ejemplo, el espejo de 73 es 37. El espejo de 135 es 531.

Procedimiento **añadirPrimos** (1 punto). Este procedimiento recibe como parámetro un número y una lista, añade a esa lista todos los números primos encontrados hasta ese número (el número 1 no cuenta). Por ejemplo, si le pasamos un 7, actualizará la lista recibida con los números 2,3,5 y 7.

Función **obtenerNumMult** (1 punto). Esta función recibe como parámetro un número, y devuelve el resultado de multiplicar todos los dígitos que la contienen. Por ejemplo, si le pasamos un 73, nos devolverá 21 (el resultado de multiplicar  $7 \cdot 3$ ). Si le pasamos el 342, devolverá 24 (el resultado de multiplicar  $3 \cdot 4 \cdot 2$ ).

**Programa principal** (1,5 puntos). Teniendo en cuenta el enunciado inicial, y apoyándote en las funciones que has desarrollado, realiza el programa principal de manera que solicite un número al usuario, y haga las comprobaciones oportunas de manera que imprima finalmente si ese número es Sheldon o no.

### Muestra de ejecución:

```
=====
PROGRAMA CÁLCULO CONJETURA SHELDON
=====
Dame el número para el que quieres comprobar
si cumple el teorema de Sheldon: 25
El número 25 NO es un número Sheldon
=====

PROGRAMA CÁLCULO CONJETURA SHELDON
=====
Dame el número para el que quieres comprobar
si cumple el teorema de Sheldon: 73
El número 73 es un número Sheldon
=====

PROGRAMA CÁLCULO CONJETURA SHELDON
=====
Dame el número para el que quieres comprobar
si cumple el teorema de Sheldon: 31
El número 31 NO es un número Sheldon
=====
```

## 2. Traductor (5 puntos).

Escribe un programa que haga de traductor de textos de castellano a inglés. Para ello, nuestro programa visualizará un **menú** (0,5 puntos) indefinidamente mostrando tres opciones:

- 1) **Introducir palabras al traductor** (2,5 puntos). El usuario introducirá las palabras en español e inglés separadas por dos puntos, y cada par palabra:traducción separados por comas. Una vez introducido nos devolverá al menú principal. Si la palabra ya existe en el traductor, nos

- informará mostrando la traducción actual en el traductor, y por tanto, no será añadida la palabra.
- 2) **Introducir texto a traducir** (2 puntos): Pedirá un texto en castellano, y realizará una traducción teniendo en cuenta las palabras existentes en el traductor. Si no existe la traducción de una palabra, ésta quedará en el idioma original (castellano).
  - 3) **Salir**: Si se selecciona esta opción el menú dejará de ejecutarse, y por tanto, finalizará el programa.

### Muestra de ejecución:

```
=====
TRADUCTOR CASTELLANO-INGLÉS =
=====
1) Introducir palabras
2) Traducir texto
3) Salir
=====
¿Qué opción deseas? 1
Introduce la lista de palabras y traducciones en formato palabra:traducción
separadas por comas: rojo:red
La palabra rojo ha sido añadida al traductor
=====
TRADUCTOR CASTELLANO-INGLÉS =
=====
1) Introducir palabras
2) Traducir texto
3) Salir
=====
¿Qué opción deseas? 1
Introduce la lista de palabras y traducciones en formato palabra:traducción
separadas por comas: roja:red,rojo:red,naranja:orange,mesa:table
La palabra roja ha sido añadida al traductor
Ojo!! La palabra rojo no ha sido añadida al traductor, ya que existe actualmente
con la traducción: red
La palabra naranja ha sido añadida al traductor
La palabra mesa ha sido añadida al traductor
=====
TRADUCTOR CASTELLANO-INGLÉS =
=====
1) Introducir palabras
2) Traducir texto
3) Salir
=====
¿Qué opción deseas? 2
Introduce una frase en español: la mesa es roja
la table es red
=====
TRADUCTOR CASTELLANO-INGLÉS =
=====
1) Introducir palabras
2) Traducir texto
3) Salir
=====
¿Qué opción deseas? 3
```