


Programación I	Examen de evaluación continua I 02/10/2022	 Deusto Facultad de Ingeniería
----------------	--	---

Nombre: _____

DNI: _____ Aula: _____ Fila: _____ Columna _____

Instrucciones

- El ejercicio a pie de máquina se entregará al terminar el examen. Se entregará en ALUD al terminar el examen..
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.
- Guarda el código de todos los ejercicios en un fichero que se llame “Examen2-NombreApellidos.py”, por ejemplo: Examen2-BorjaSanz.py.

1. Ejercicio (1,5 puntos)

Crear una función `contar_segundos()` que recibe 3 parámetros enteros: horas, minutos y segundos y el número total de segundos.

Comprueba que funciona correctamente con varios ejemplos:

```
print(contarSegundos(1,20,30)) # Debería devolver 4830
print(contarSegundos(0,0,15)) # Debería devolver 15
```

2. Ejercicio (2,5 puntos)


Crear una función `separar_mayores()` que reciba dos parámetros: una lista de enteros y un número entero y devuelva una lista con todos aquellos valores de la lista que sean mayores que el segundo parámetro.

Comprueba que funciona correctamente con varios ejemplos:

```
print(separarMayores([1,2,3,4,5,6], 4)) # Debería devolver [4,5,6]
print(separarMayores([4,5,6,1,3]), 3) # Debería devolver [4,5,6,3]
```

3. Ejercicio (4 puntos)

Programa una función `login()` que reciba el número de intentos posibles para hacer login (entero). La función debe de solicitar el usuario y la contraseña al usuario en cada intento y devuelva

Programación I	Examen de evaluación continua I 02/10/2022	 Deusto Facultad de Ingeniería
----------------	--	---

Verdadero si el usuario es igual a “user1” y la contraseña a “pass1”; y deberá devolver Falso si el usuario y contraseña no han sido correctos en el número máximo de intentos indicado.

Introduce el usuario: user2

Introduce la contraseña: pass1

Datos incorrectos.

Número de intentos 1

Introduce el usuario: user1

Introduce la contraseña: pass1

Número de intentos 2 # Devuelve True

4. Ejercicio (4 puntos)

Programa una función `ganador` que reciba como parámetros una lista con el piloto y con tiempos de carrera (expresados en horas, minutos y segundos) del siguiente tipo:

```
[["Piloto1", [1, 20, 10]],
 ["Piloto2", [1, 25, 13]],
 ["Piloto3", [1, 19, 18]]]
```

Y devuelva el registro del ganador, es decir que menos ha tardado.

Comprueba que la función funciona correctamente con varios ejemplos:

```
print(ganador(lista_tiempos)) # Devuelve ["Piloto3", [1, 19, 18]]
```