



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE PROGRAMACIÓN I

23-OCTUBRE-2023

Nombre: _____

DNI: _____ Aula: _____ Fila: _____ Columna _____

Instrucciones

Confidential

- El ejercicio a pie de máquina se entregará 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 cada ejercicio en un fichero que se llame "EjerX-NombreApellidos.py", por ejemplo: Ejer1-MariluzGuenaga.py. Comprime todos los ficheros para su entrega.
- El ejercicio debe incluir un comentario al inicio con el nombre, apellidos y DNI del alumno o alumna.

1. Ejercicio (1 punto)

Programa una función `ultimaContiene()` que reciba una lista de palabras y un número `k`, y devuelve la última palabra con un número de letras mayor o igual a `k`. Siempre habrá palabras con `k` o más caracteres.

Comprueba que la función funciona correctamente con varios ejemplos:

```
lista = ["perro", "gato", "ratón", "elefante", "avestruz", "sardina", "rata", "atún"]
print(ultimaContiene(lista, 5)) #Debe mostrar sardina
print(ultimaContiene(lista, 8)) #Debe mostrar avestruz
```

2. Ejercicio (2 puntos)

Programa una función `multiplos()` que reciba una lista de enteros y un número `k` y devuelva una nueva lista con todos los valores de la lista que son múltiplos de `k`. Comprueba que la función funciona correctamente con varios ejemplos:

```
lista = [1, 2, 6, 54, 25, 17, 10, 3]
print(multiplos(lista, 3)) #[6, 54, 3]
```

3. Ejercicio (2 puntos)

Programa una función `mover_jugador()` para mover al jugador de un juego según una cadena encriptada que se recibe por parámetro. Los caracteres de la cadena determinarán los movimientos del jugador en los ejes X e Y, según la siguiente codificación:

- Cada vocal (mayúscula o minúscula) supone un paso hacia el norte.
- Cada dígito supone un paso hacia el sur
- Cada consonante mayúscula supone un paso hacia el este
- Cada consonante minúscula supone un paso hacia el oeste
- El resto de los caracteres implican ganar una moneda

La función retorna la posición y el número de monedas

Comprueba que funciona correctamente con varios ejemplos:

```
posX, posY, monedas = mover_jugador("La nota: 9.5 ¡sobre!")
print(f"El jugador se ha movido a la posición ({posX}, {posY}) y ha conseguido {monedas} monedas")
```

El jugador se ha movido a la posición (-4 , 3) y ha conseguido 7 monedas

```
posX, posY, monedas = mover_jugador("Paris-3.56")
print(f"El jugador se ha movido a la posición ({posX}, {posY}) y ha conseguido {monedas} monedas")
```

El jugador se ha movido a la posición (-1 , -1) y ha conseguido 2 monedas



Confidential

4. Ejercicio (3 puntos)

Programa una función `mayor_densidad()` que recibe una lista con la información de ciudades (tal y como se muestra a continuación) y devuelve la población con mayor densidad de población. Por cada ciudad, tenemos su nombre, población (habitantes) y km2. Densidad de población = número de habitantes / superficie en km2

```
poblaciones = [  
    ["Durango", 28226, 10.79],  
    ["Arrigorriaga", 12160, 16.36],  
    ["Otxandio", 1296, 12.43],  
    ["Eibar", 27282, 24.78],  
    ["Zarautz", 23101, 14.80]  
]  
  
print(mayor_densidad(poblaciones))  
# Debería devolver ("Durango", 2615.94)  
# Esto es, la población con mayor densidad es Durango que tiene 2615.94 hab/km2
```