

Introducción a la Programación

2ª Prueba de Evaluación continua

2023-2024

DNI:

NOMBRE Y APELLIDOS:

Se pide crear una aplicación que ayude a la gestión de los repartos que realiza una empresa de paquetería.

De cada reparto se guarda la siguiente información:

- **id:** identificador del reparto
- **cod_repartidor:** código del repartidor
- **pedidos:** lista que contiene los códigos de los pedidos transportados en ese reparto

Los datos de los repartos serán los siguientes:

```
repartos = [{ 'id': 'C03', 'cod_repartidor': '125', 'pedidos': [5, 8, 7, 6] },  
             { 'id': 'C04', 'cod_repartidor': '126', 'pedidos': [4, 10, 2] },  
             { 'id': 'C05', 'cod_repartidor': '127', 'pedidos': [1, 11, 12, 14] },  
             { 'id': 'C06', 'cod_repartidor': '125', 'pedidos': [15, 16, 17, 20] },  
             { 'id': 'C07', 'cod_repartidor': '126', 'pedidos': [3, 21, 23, 24] },  
             { 'id': 'C08', 'cod_repartidor': '125', 'pedidos': [25, 26] }]
```

Partiendo de esta información se pide que implementéis las siguientes funciones en un único fichero denominado **examen.py**:

1. **obtenerMediaPedidos:** función que recibe la lista de repartos y devuelve la media de pedidos transportados en cada reparto (número de pedidos / número de repartos). **(0,5 puntos)**

```
print(obtenerMediaPedidos(repartos))#Devuelve 3.5
```

2. **calcularSueldo:** función que recibe el código de un repartidor y la lista de repartos y devuelve el sueldo para ese repartidor. Para calcularlo debes tener en cuenta que, por cada reparto realizado, si el número de pedidos es menor o igual a 3, recibirá 8€ y por aquellos repartos con más de 3 pedidos, recibirá 15€. **(0,5 puntos)**

```
print(calcularSueldo('125', repartos))#Devuelve 38
```

```
print(calcularSueldo('126', repartos))#Devuelve 23
```

3. **repartosPorRepartidor**: función que recibe la lista de repartos y devuelve un diccionario en el que se indicará por cada repartidor el número de repartos que ha realizado. Las claves de este diccionario serán los códigos de los repartidores y los valores serán el número de repartos realizados. **(0,7 puntos)**:

```
repartosPorRepartidor(repartos)#Devuelve {'125': 3, '126': 2, '127': 1}
```

4. **repartidorMasRepartos**: función que recibe el diccionario del ejercicio anterior y devuelve el código del repartidor que más repartos ha realizado. En caso de que no hayas conseguido el ejercicio anterior, puedes copiar el diccionario anterior. **(0,6 puntos)**

Si te ha salido el ejercicio anterior:

```
print(repartidorMasRepartos(repartosPorRepartidor(repartos)))
```

En caso de que no te haya salido el ejercicio anterior:

```
rep_repartidor= {'125': 3, '126': 2, '127': 1}  
print(repartidorMasRepartos(rep_repartidor))
```

5. **guardarRepartos**: función que recibe la lista de repartos y guarde los datos de la lista en el fichero repartos.csv con el siguiente formato: **(0,7 puntos)**

```
C03;125;5;8;7;6
```

```
C04;126;4;10;2
```

```
C05;127;1;11;12;14
```

```
C06;125;15;16;17;20
```

```
C07;126;3;21;23;24
```

```
C08;125;25;26
```

```
guardarRepartos(repartos)
```