



--	--	--	--	--

APELLIDOS: _____ NOMBRE: _____

DNI: _____ TITULACIÓN: ☐ IC ☐ IS ☐ TI ☐ DG GRUPO: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Disponemos de datos sobre los entrenamientos realizados por un usuario. Para cada entrenamiento se tiene la siguiente información:

- **tipo**: tipo de entrenamiento realizado, de tipo str.
- **fechahora**: fecha y hora del entrenamiento, de tipo datetime.
- **ubicación**: lugar donde se ha realizado el entrenamiento, de tipo str
- **duración**: duración del entrenamiento en minutos, de tipo int.
- **calorías**: número de kilocalorías activas quemadas en el entrenamiento, de tipo int.
- **distancia**: distancia en kilómetros recorrida en el entrenamiento, de tipo float.
- **frecuencia**: frecuencia cardiaca media registrada en el entrenamiento, de tipo int.
- **compartido**: indica si el entrenamiento ha sido compartido por el usuario, de tipo bool (S si ha sido compartido, N si no lo ha sido).

Por ejemplo, la siguiente línea del fichero:

Andar,12/11/2021 8:14,Sevilla,48,155,3.49,89,N

indica que el 12 de noviembre de 2021 a las 8:14 horas, el usuario realizó un entrenamiento de tipo 'Andar' durante 48 minutos, quemando 155 kilocalorías activas y recorriendo un total de 3.49 kilómetros con una frecuencia cardiaca media de 89 lpm, y que no compartió este entreno.

Para almacenar los datos de un entrenamiento se usará obligatoriamente la siguiente namedtuple:

```
Entreno = namedtuple('Entreno', ('tipo', 'fechahora', 'ubicacion', 'duracion',  
                                'calorias', 'distancia', 'frecuencia', 'compartido'))
```

Cree un módulo **entrenos.py** e implemente en él las funciones que se piden. Cree un módulo **entrenos_test.py** y defina una función de test para cada función solicitada. Puede definir funciones auxiliares cuando lo considere necesario:

1. **lee_entrenos**: lee un fichero de entrada en formato CSV codificado en UTF-8 y devuelve una lista de tuplas de tipo Entreno conteniendo todos los datos almacenados en el fichero. Utilice la función `datetime.strptime` con el formato `"%d/%m/%Y %H:%M"` para convertir la fecha. (1 punto)
2. **porcentaje_calorias_por_tipo**: recibe una lista de tuplas de tipo Entreno y un conjunto de tipos de entrenamientos, y devuelve un diccionario que relaciona cada tipo de entrenamiento presente en el conjunto de entrada con el porcentaje de calorías quemadas en ese tipo de entrenamiento respecto al total de calorías quemadas en todos los entrenamientos. (2 puntos)
Por ejemplo, si el usuario ha realizado entrenos de los tipos 'Andar', 'Bici' y 'Correr', las calorías quemadas en cada uno de ellos han sido 1000, 2000 y 7000, respectivamente, y la lista de entrada es ['Bici', 'Correr'], el diccionario devuelto debe ser {'Bici': 20.0, 'Correr': 70.0}. TEST
3. **año_mayor_distancia_media**: recibe una lista de tuplas de tipo Entreno, un valor booleano `c` y un valor entero `d`, y devuelve el año con la mayor distancia media de todos los entrenamientos del año cuya propiedad 'compartido' tenga el valor `c` y cuya duración sea superior al valor `d`. Si el valor de `c` es `None`, se tendrán en cuenta tanto los entrenamientos compartidos como los no compartidos. (2 puntos)
Por ejemplo, si `c` tiene el valor `None` y `d` el valor 5.0, el usuario ha realizado en el año 2020 tres entrenamientos con distancias 40, 15 y 20, y en el año 2021 dos entrenamientos con distancias 40 y 20, el valor devuelto debe ser 2021, ya que la media es de 30, por 25 del año 2020.

4. **entrenos_mas_repetidos**: recibe una lista de tuplas de tipo Entreno y dos valores de tipo date f1 y f2, y devuelve un diccionario que relaciona cada ubicación con el tipo de entrenamiento que más veces se ha realizado en esa ubicación, para los entrenamientos realizados entre las fechas f1 y f2 (ambas inclusive). (2,5 puntos)

Por ejemplo, si el usuario ha realizado en la ubicación 'Sevilla' 25 entrenamientos de tipo 'Andar', 10 de tipo 'Correr' y 20 de tipo 'Bici' entre las dos fechas recibidas como parámetros, el valor asociado a la clave 'Sevilla' del diccionario debe ser 'Andar'.

5. **incrementos_anuales_distancia**: recibe una lista de tuplas de tipo Entreno, y devuelve una lista con los incrementos en las distancias recorridas en los entrenamientos de un año respecto al anterior. Nota: es posible que en algún año no haya habido ningún entrenamiento. (2,5 puntos)

Por ejemplo, si los años 2019, 2021 y 2022 se recorrieron 920.0, 750.0, y 1010.0 kms respectivamente, y el año 2020 no se realizaron entrenamientos, la lista devuelta debe ser [-920.0, 750.0, 260.0].