

 FUNDAMENTOS DE PROGRAMACIÓN. Curso 2021/22 PRIMER EXAMEN PARCIAL. 11 de febrero de 2022	Código (5 primeros caracteres):				
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
APELLIDOS: _____ NOMBRE: _____ DNI: _____ TITULACIÓN: <input type="checkbox"/> IC <input type="checkbox"/> IS <input type="checkbox"/> TI <input type="checkbox"/> DG GRUPO: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5					

Disponemos de datos sobre las rutas en moto realizados por un motorista experimentado en la provincia de Sevilla. Para cada ruta se tiene la siguiente información:

- **ciudad_inicio**: ciudad desde donde el usuario comenzó la ruta, de tipo str.
- **latitud**: latitud de la ciudad de inicio, de tipo float.
- **longitud**: longitud de la ciudad de inicio, de tipo float.
- **fecha_ruta**: día en el que se realizó la ruta, de tipo date.
- **km**: número de kilómetros recorridos en la ruta, de tipo float.
- **gasolineras**: número de gasolineras que se encontró mientras hacía la ruta, de tipo int.
- **dificultad**: dificultad de la ruta, este valor toma los valores de baja, media o alta, de tipo str.
- **zona_descanso**: indica si en la ruta existen zonas de descanso, de tipo bool.
- **vel_max**: velocidad máxima que ha alcanzado a lo largo de todo el trayecto de la ruta, expresada en km/h, de tipo int.
- **vel_min**: velocidad mínima que ha alcanzado a lo largo de todo el trayecto de la ruta, expresada en km/h, de tipo int.

Por ejemplo, la siguiente línea del fichero:

```
Aguadulce;37.25/-4.991359;8/8/2021;32.21;9;alta;True;106;66
```

indica que el 8 de agosto de 2021, el usuario realizó una ruta que comenzó en Aguadulce, cuyas coordenadas en el mapa son 37.25 y -4.991359, con un total de 32.21 kilómetros recorridos. En el transcurso de la ruta se encontró con un total de 9 gasolineras, considerándola una ruta con dificultad alta. Había al menos alguna zona de descanso, y las velocidades máxima y mínima alcanzadas fueron 106 y 66, respectivamente.

Para almacenar los datos de una ruta se usará obligatoriamente la siguiente namedtuple:

```
Ruta = namedtuple('Ruta', 'ciudad_inicio, coordenada, fecha_ruta,
                           km, gasolineras, dificultad, zona_descanso, vel_max, vel_min')
```

donde 'coordenada' es otra namedtuple que guarda la información de la latitud y la longitud, tal como se define a continuación:

```
Coordenada = namedtuple('Coordenada', 'latitud, longitud')
```

Cree un módulo **rutas.py** e implemente en él las funciones que se piden. Puede definir funciones auxiliares cuando lo considere necesario. Además, también debe crear un módulo **test_rutas.py** en el que pruebe todas las funciones anteriores. Para ello, escriba en primer lugar una llamada para cargar los datos del fichero CSV y muestre dichos datos. A continuación, escriba llamadas al resto de funciones y muestre los resultados devueltos por ellas.

1. **lee_rutas**: lee un fichero de entrada en formato CSV codificado en UTF-8, y devuelve una lista de tuplas de tipo `Ruta` conteniendo todos los datos almacenados en el fichero. Utilice la función `datetime.strptime(cadena, formato).date()` con el formato `'%m/%d/%Y'` para convertir la fecha. Tenga en cuenta que hay valores en el csv para la fecha que están vacíos (su valor es igual a `""`), en este caso deberá asignarle el valor del día actual (`datetime.today().date()`). Use el método de cadenas `split` para separar la latitud de la longitud a la hora de hacer la lectura de la coordenada. Por último, tenga en cuenta que el atributo `ciudad_inicio` puede venir con un espacio adicional al final que hay que eliminar; use la función `strip` para ello. (1 pto)

2. **acumular_kms_por_meses**: recibe una lista de tuplas de tipo Ruta, y devuelve un diccionario que asocia a cada año una lista con el total de kilómetros que se han recorrido en cada mes. Si en alguno de los meses no se ha realizado ninguna ruta, debe aparecer el valor 0.0. Los valores en la lista deben estar ordenados cronológicamente, es decir, el primer valor se corresponde con enero, el segundo con febrero, y así sucesivamente. (3 pts)

Por ejemplo, con los datos del fichero, el valor devuelto debe ser:

```
{2021: [426.04, 514.24, 1060.5, 863.1, 507.82, 672.4, 962.27, 774.1, 798.38, 918.56, 410.67, 370.63],
 2022: [604.88, 1849.68, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
```

indicando que, en enero del 2022, el total de kilómetros recorridos por el motorista fueron 604.88.

3. **diferencias_kms_meses_anyo**: recibe una lista de tuplas de tipo Ruta, y devuelve un diccionario que asocia cada año con una lista con las diferencias en kilómetros recorridos de cada mes de ese año con respecto al mes anterior. Las diferencias deben estar ordenadas cronológicamente, es decir, el primer valor es la diferencia entre febrero y enero, el segundo la diferencia entre marzo y febrero, y así sucesivamente. Tiene que usar de forma obligatoria la función del apartado anterior. (2 pts)

Por ejemplo, con los datos del fichero, el valor devuelto debe ser:

```
{2021: [88.2, 546.26, -197.41, -355.27, 164.58, 289.87, -188.18, 24.29, 120.18, -507.89, -40.04],
 2022: [1244.8, -1849.68, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
```

indicando que, en el año 2021, la diferencia de kilómetros totales recorridos entre febrero y enero fue 88.2.

4. **top_rutas_lejanas**: Dada una lista de tuplas de tipo Ruta, un valor entero n, un valor c de tipo Coordenada, y un entero km_min, obtener una lista con las n rutas cuya ciudad de inicio está más lejana a las coordenadas que se pasan como parámetro de entrada y cuyo número de kilómetros sea mayor al valor km_min. La variable km_min tomará como valor por defecto None, en cuyo caso se tendrán en cuenta todas las rutas. Para calcular la distancia entre las distintas ciudades, deberá usar la distancia Manhattan. Dadas dos coordenadas c1, y c2, la distancia Manhattan se calcula como $d = |lat1-lat2| + |long1-long2|$. Donde lat1 y long1 son la latitud y longitud de c1, y lat2 y long2 son la latitud y longitud de c2. Use el método *abs* para obtener el valor absoluto. (1,5 pts)

Por ejemplo, si n tiene el valor 2, c el valor (35.15, -8.76) y km_min = None, el valor devuelto debe ser:

```
[Ruta(ciudad_inicio='Badolatosa', coordenada=Coordenada(latitud=37.31, longitud=-4.674225),
 fecha_ruta=datetime.date(2021, 10, 20), km=148.07, gasolineras=2, dificultad='baja',
 zona_descanso=False, vel_max=127, vel_min=87),
 Ruta(ciudad_inicio='Casariche', coordenada=Coordenada(latitud=37.29, longitud=-4.759777),
 fecha_ruta=datetime.date(2021, 2, 26), km=93.53, gasolineras=10, dificultad='baja',
 zona_descanso=False, vel_max=118, vel_min=78)]
```

indicando que Badolatosa y Casariche son las ciudades más lejanas a las coordenadas 35.15 y -8.76. Para hacerse una idea de si está haciendo bien o no el ejercicio, puede usar las coordenadas específicas de Sevilla, que son lat = 37.3826 y long = -5.99629.

5. **ciudades_top_tiempo_dificultad**: Dada una lista de tuplas de tipo Ruta y un valor entero n, obtener un diccionario que relacione cada dificultad con las ciudades de inicio de las n rutas con zona de descanso que han tardado más tiempo en hacerse, ordenadas de mayor a menor tiempo. Si suponemos que la velocidad de las rutas ha sido siempre constante y con valor vel_min, podemos calcular el tiempo usando la fórmula $t = km/vel_min$. El parámetro n tendrá un valor por defecto igual a 3. (2,5 pts)

Por ejemplo, si el valor de n es 3, el valor devuelto debe ser:

```
{alta: ['Montellano', 'Aguadulce', 'Guadalcanal'],
 media: ['Pedroso (El)', 'Pilas', 'Gines'], baja: ['Rinconada (La)', 'Tocina', 'Algámitas']}
```

indicando que, para la dificultad alta, las 3 rutas que más tiempo han tardado en hacerse han sido las que empezaron en Montellano, Aguadulce y Guadalcanal.