



Tenemos un conjunto de partidos de baloncesto disputados a cuatro cuartos de forma que gana el partido el equipo que más puntos consigue sumando las cuatro partes. Los datos tienen esta forma en un fichero csv:

```
12/10/2019;Barcelona;Zaragoza;Copa del Rey;18-20*15-18*24-17*20-17;24;12
23/11/2018;Iberostar Tenerife;Unicaja;Eurocopa;19-23*15-22*20-24*19-16;13;15
```

La información de cada línea se corresponde con lo siguiente:

- Fecha del partido.
- Primer equipo participante.
- Segundo equipo participante.
- Competición a la que pertenece el partido.
- Resultados de los cuatro cuartos. Cada cuarto viene separado por el carácter `*`, y contiene los puntos anotados por el primer y el segundo equipo, respectivamente, separados por el carácter `.`.
- Faltas personales cometidas por el primer equipo.
- Faltas personales cometidas por el segundo equipo.

Por ejemplo, en la primera línea, vemos que se enfrentaron el Barcelona y el Zaragoza, dentro de un partido de la Copa del Rey, disputado el 12 de octubre de 2019. El Barcelona anotó 18 puntos en el primer cuarto, y el Zaragoza 20; en el segundo cuarto, anotaron 15 y 18 puntos, y así sucesivamente. Sumando los puntos de cada cuarto, obtendríamos que el Barcelona ganó el partido 77 a 72. Finalmente, el Barcelona cometió 24 faltas personales y el Zaragoza 12.

Usaremos esta definición de namedtuple:

```
PartidoBasket = namedtuple('PartidoBasket', 'fecha, equipo1, equipo2, competicion, puntos_eq1, puntos_eq2, faltas_eq1, faltas_eq2')
```

Nótese que almacenaremos **el total de puntos anotados por cada equipo** en todo el partido. Por ejemplo, para la primera línea del csv, obtendremos la siguiente tupla (fíjese en el tipo de cada uno de los campos):

```
PartidoBasket(fecha=datetime.date(2019, 10, 12), equipo1='Barcelona', equipo2='Zaragoza', competicion='Copa del Rey', puntos_eq1=77, puntos_eq2=72, faltas_eq1=24, faltas_eq2=12)
```

Implemente las siguientes funciones en un módulo `baloncesto.py`:

1. **parsea_y_suma_resultados**: recibe una cadena de texto con los resultados de los cuatro cuartos de un partido, y devuelve una tupla con dos enteros, correspondientes a los puntos totales anotados por el primer y el segundo equipo. Por ejemplo, si recibe la cadena '18-20*15-18*24-17*20-17', debe devolver (77, 72). (0,5 puntos)
2. **lee_partidos**: recibe una cadena de texto con la ruta de un fichero csv, y devuelve una lista de tuplas `PartidoBasket` con la información contenida en el fichero. Utilice `datetime.strptime(cadena, "%d/%m/%Y").date()` para parsear las fechas. Use la función `parsea_y_suma_resultados` para obtener los puntos totales de los equipos. (1 punto)
3. **equipo_con_mas_faltas**: recibe una lista de tuplas `PartidoBasket`, y un conjunto de cadenas de texto equipos, con valor por defecto `None`, y devuelve el nombre del equipo que acumula más faltas personales, de entre los equipos incluidos en el parámetro equipos. Si el parámetro equipos es `None`, se devolverá el equipo con más faltas personales de entre todos los que aparezcan en la lista de partidos recibida. (1,5 puntos)

4. **media_puntos_por_equipo:** recibe una lista de tuplas PartidoBasket y una cadena de texto competicion, y devuelve un diccionario en el que se relaciona cada equipo con la media de puntos anotados por el equipo en todos los partidos disputados de la competición indicada por el parámetro competicion. (1,5 puntos)
5. **diferencia_puntos_anotados:** recibe una lista de tuplas PartidoBasket y una cadena de texto equipo, y devuelve una lista de enteros con la diferencia de puntos anotados entre cada dos partidos consecutivos del equipo indicado por el parámetro equipo. Por ejemplo, si el equipo indicado ha jugado tres partidos consecutivos en el tiempo, anotando 60, 64 y 58 respectivamente, la lista devuelta debería ser [4, -6]. Tenga en cuenta que los partidos no tienen por qué venir ordenados cronológicamente en la lista de tuplas recibida. (1,5 puntos)

Las tres siguientes funciones están relacionadas entre sí, de manera que cada una se debe utilizar para resolver la siguiente:

6. **victorias_por_equipo:** recibe una lista de tuplas PartidoBasket y devuelve un diccionario que hace corresponder cada equipo con el número de victorias del mismo. (1 punto)
 7. **equipos_minimo_victorias:** recibe una lista de tuplas PartidoBasket y un entero n, y devuelve una lista con los equipos con n o más victorias. La lista estará ordenada de mayor a menor número de victorias. (1 puntos)
 8. **equipos_mas_victorias_por_año:** recibe una lista de tuplas PartidoBasket y un entero n, y devuelve un diccionario en el que las claves son los años en los que se han disputado los partidos, y los valores son listas con los equipos con n o más victorias acumuladas en los partidos disputados cada año. Las listas estarán ordenadas de mayor a menor número de victorias. (1 punto)
9. Pruebe las funciones implementadas en un módulo baloncesto_test.py. Se recomienda que lo vaya haciendo a medida que vaya resolviendo los distintos apartados. (1 punto)

Anexo: Pruebas de las funciones.

En esta sección, se muestra una posible ejecución de las pruebas de las funciones.

```
Test de lee_datos_baloncesto:
Total registros leídos: 200
Mostrando los tres primeros registros:
    PartidoBasket(fecha=datetime.date(2019, 10, 12), equipo1='Barcelona',
equipo2='Zaragoza', competicion='Copa del Rey', puntos_eq1=77, puntos_eq2=72,
faltas_eq1=24, faltas_eq2=12)
    PartidoBasket(fecha=datetime.date(2018, 11, 23), equipo1='Iberostar
Tenerife', equipo2='Unicaja', competicion='Eurocopa', puntos_eq1=73,
puntos_eq2=85, faltas_eq1=13, faltas_eq2=15)
    PartidoBasket(fecha=datetime.date(2018, 5, 12), equipo1='Joventut
Badalona', equipo2='Valencia Basket', competicion='Super Copa', puntos_eq1=86,
puntos_eq2=81, faltas_eq1=20, faltas_eq2=21)
```

Test de equipo_mas_faltas:

('Gran Canaria', 591)

Test de media_puntos_por_equipo (competicion='Copa del Rey'):

{'Barcelona': 83.75, 'Zaragoza': 78.16666666666667, 'Monbus Obradoiro': 77.57142857142857, 'Estudiantes': 76.55555555555556, 'MoraBanc Andorra': 77.7, 'Unicaja': 83.16666666666667, 'Gran Canaria': 83.88888888888889, 'Valencia Basket': 80.0, 'Baskonia': 79.81818181818181, 'Real Madrid': 81.6, 'Joventut Badalona': 78.57142857142857, 'Bilbao Basket': 86.8, 'Fuenlabrada': 76.8, 'Tofas Bursa': 83.4, 'San Pablo Burgos': 84.25, 'Casademont Zaragoza': 79.5, 'Gipuzkoa Basket': 75.0, 'Iberostar Tenerife': 82.5}

Test de diferencia_puntos_anotados (equipo='Barcelona')

[-7, 11, -14, 13, -5, -2, 21, -15, -14, 20, -4, 8, -12, 0, 2, 0, -5, 6, -15, 16, 3, -15, 16, -9, 4, 3, -10]

Test de victorias_por_equipo

Counter({'Barcelona': 18, 'San Pablo Burgos': 18, 'Bilbao Basket': 17, 'Gran Canaria': 17, 'Zaragoza': 14, 'MoraBanc Andorra': 13, 'Unicaja': 13, 'Monbus Obradoiro': 11, 'Baskonia': 11, 'Real Madrid': 10, 'Joventut Badalona': 9, 'Valencia Basket': 8, 'Fuenlabrada': 8, 'Casademont Zaragoza': 7, 'Estudiantes': 6, 'Tofas Bursa': 6, 'Iberostar Tenerife': 3, 'Gipuzkoa Basket': 1})

Test de equipos_minimo_victorias (n=8)

['Barcelona', 'San Pablo Burgos', 'Bilbao Basket', 'Gran Canaria', 'Zaragoza', 'MoraBanc Andorra', 'Unicaja', 'Monbus Obradoiro', 'Baskonia', 'Real Madrid', 'Joventut Badalona', 'Valencia Basket', 'Fuenlabrada']

Test de equipos_mas_victorias_por_año (n=8)

{2019: [], 2018: ['Barcelona'], 2020: ['Bilbao Basket', 'Gran Canaria']}