

Programación—Certamen Recuperativo - 27 de Noviembre de 2014

Nombre: Rol: -

2. [35 %] Un centro vacacional interesado en la entretención de sus veraneantes, guarda su información en un **diccionario**, cuya llave es un código interno del turista, y el valor es una tupla que contiene: nombre, código entretenimiento, cantidad de días a la semana que asiste.

```
turistas = {
    102: ('Aguiles Bailo', 6, 3),
    23 : ('Alan Brito', 11, 4),
    55 : ('Armando Casas', 5, 4),
    1200 : ('Mario Neta', 7, 2),
    755: ('Elmer Curio', 7, 5),
    521: ('Luz Rojas', 2, 3)
    # ...
}
```

Por ejemplo, el turista 521 tiene nombre Luz Rojas, participa en la entretención 2, y asiste 3 veces a la semana. Considere que el diccionario puede tener muchas personas, no sólo los del ejemplo.

Por otro lado, existe una **lista** de tuplas con las entretenciones del centro vacacional y una clasificación (de 1 a 7) entregada por los usuarios de acuerdo a su grado de conformidad con ella. El índice de la lista representa el código entretenimiento. Por ejemplo, la primera entretención (código 0) es Spa.

```
entretencion = [('Spa',5), ('Masajes',6), ('Casino',3), ('Piscina',7),
                ('Playa',4), ('Surf',7), ('Baile entretenido',2), ('Bar',1), ('
                Restaurant',6), ('Karaoke',4), ('Bingo',7), ('Cine',5), ('Teatro',5)
                , ('Buceo',5), ('Gimnasio',4)]
```

- a) Desarrolle la función `filtrar(entretencion, nota)` que reciba como parámetro la **lista** entretención y un entero. La función debe retornar una **lista** con las entretenciones cuya clasificación sea mayor o igual a la nota pasada por parámetro.

```
>>> filtrar(entretencion, 6)
['Masajes', 'Piscina', 'Surf', 'Restaurant', 'Bingo']
```

- b) Desarrolle la función `promedio_dias(turistas, entretencion, e)` que reciba como parámetro el **diccionario** `turistas`, la **lista** entretención y un string con el nombre de una entretención en particular. La función debe retornar el promedio de días que acuden las personas a dicha entretención.

```
>>> promedio_dias(turistas, entretencion, 'Bar')
3.5
```

- c) Desarrolle la función `usuarios_por_entretencion(turistas, entretencion)`, que reciba como parámetro el **diccionario** `turistas` y la **lista** entretención. Esta función debe retornar un **diccionario** que asocie el nombre de la entretención a una lista de códigos de turistas que asiste a ella. La lista debe estar ordenada en forma ascendente. Si la entretención no tiene usuarios, no la incluya en el diccionario.

```
>>> usuarios_por_entretencion (turistas, entretencion)
{'Baile entretenido': [102], 'Cine':[23], 'Surf': [55], 'Bar': [755,
1200], 'Casino': [521]}
```

Programación—Certamen Recuperativo - 27 de Noviembre de 2014

Nombre:

Rol: -

3. [40 %] La automotora PyCar requiere generar información importante con respecto a sus ventas. Principalmente requiere realizar un análisis sobre el desempeño de sus vendedores así como también sobre la evolución de las ventas que se van realizando.

Para lograr este propósito, PyCar tiene almacenado en un archivo del tipo `productos.txt` la información del precio de los automóviles que tiene a la venta. Cada producto viene especificado por la marca del automóvil, el modelo y el precio de venta. Por otro lado, PyCar almacena en un archivo del tipo `ventas.txt`, cada una de las transacciones realizadas por concepto de la venta de un automóvil (ordenadas por fecha), esta información cuenta con el día de la venta, la hora en que se ha realizado, el modelo del automóvil, la marca del mismo y el código del vendedor que ha realizado la transacción.

Nota: los archivos son ejemplos, su programa debe funcionar para toda la cantidad de automóviles que puede tener la automotora y para todas las ventas que se han realizado durante el año 2014.

`productos.txt`

```
hyundai,accent,7390000
hyundai,tucson,11690000
hyundai,santafe,15790000
suzuki,nomade,9990000
suzuki,vitara,9490000
suzuki,swift,7190000
toyota,yaris,7890000
toyota,corolla,9390000
toyota,rav4,12090000
```

`ventas.txt`

```
2014/9/30,19:45,suzuki,vitara,5678
2014/9/30,20:15,toyota,corolla,4523
2014/9/30,10:40,suzuki,nomade,6547
2014/9/30,14:40,suzuki,vitara,5678
2014/10/1,10:40,toyota,yaris,4523
2014/10/1,14:40,suzuki,vitara,6547
2014/10/2,10:40,hyundai,accent,6547
2014/10/2,12:40,suzuki,vitara,5678
2014/10/2,13:20,hyundai,santafe,4523
```

Con el fin de ayudar a PyCar en sus requerimientos, Ud. debe realizar lo siguiente:

- a) Desarrolle la función `precios(archivo)` que recibe el nombre del archivo con la información de los productos y retorne un **diccionario**, donde la llave sea el modelo del automóvil y el valor sea el precio del mismo.

```
>>> precios('productos.txt')
{'rav4': 12090000, 'corolla': 9390000, 'accent': 7390000, 'nomade':
 9990000, 'yaris': 7890000, 'tucson': 11690000, 'santafe': 15790000,
 'swift': 7190000, 'vitara': 9490000}
```

- b) Desarrolle la función `total_ventas_mes(codigo, mes, archivo1, archivo2)` que recibe el código de un vendedor, un mes en particular, el nombre del archivo con la información de los productos y el nombre del archivo con la información de las ventas y retorne el **total** de ventas que ha realizado el vendedor en ese mes.

```
>>> total_ventas_mes(4523, 10, 'productos.txt', 'ventas.txt')
23680000
```

- c) Desarrolle la función `venta_diaria(archivo1, archivo2)` que recibe el nombre del archivo con la información de los productos y el nombre del archivo con la información de las ventas y genere el archivo `venta_diaria.txt` indicando la venta total que se ha realizado por día.

```
>>> venta_diaria('productos.txt', 'ventas.txt')
>>>
```

`venta_diaria.txt`

```
2014/9/30:38360000
2014/10/1:17380000
2014/10/2:32670000
```