

Programación—Certamen 2 - Jueves 1 de Agosto de 2013

Nombre:

Rol: -

2. [35%] La tienda Pylabella maneja la información de sus clientes en un diccionario cuya clave corresponde al RUT del cliente y su valor a una lista con su información. Esta información está compuesta por el nombre del cliente, su edad, la sucursal donde realizó su inscripción, y una tupla con los montos de las tres últimas compras. Considere la siguiente estructura **como ejemplo**:

```
clientes = {
    #RUT: [nombre, edad, sucursal, (monto1, monto2, monto3)]
    '16034124-5': ('Fernando Ruiz Diaz', 30, 'Vina', (100, 200, 300)),
    '5436576-2': ('Mike Portnoy', 20, 'Quilpue', (100, 100, 50)),
    '3333333-3': ('Slash', 50, 'Vina', (500, 550, 300)),
    '1234567-8': ('Cliff Burton', 35, 'Valparaiso', (50, 50, 100))
    #...
}
```

- a) Escriba la función `misma_sucursal(RUT1, RUT2, clientes)` que reciba el RUT de dos clientes, RUT1 y RUT2, además del diccionario `clientes` y retorne **True** en caso de que los clientes se inscribieron en la misma sucursal o **False** en caso contrario.:

```
>>> misma_sucursal('16034124-5', '3333333-3', clientes)
True
>>> misma_sucursal('5436576-2', '3333333-3', clientes)
False
```

- b) Escriba la función `mayores_que(edad_minima, clientes)`, que devuelva una lista con los nombres de todos los clientes cuya edad sea mayor o igual al parámetro `edad_minima`.

```
>>> mayores_que(50, clientes)
['Slash']
>>> mayores_que(25, clientes)
['Fernando Ruiz Diaz', 'Cliff Burton', 'Slash']
```

- c) Escriba la función `es_cliente_vip(RUT, monto_compra, clientes)`, la cual recibe como parámetro el RUT de un cliente, el `monto_compra` que representa el monto mínimo para ser considerado cliente VIP y el diccionario `clientes`. Esta función debe retornar **True** si la suma de las 3 últimas compras del cliente es superior o igual al `monto_compra`, y **False** en caso contrario.

```
>>> es_cliente_vip('5436576-2', 500, clientes)
False
>>> es_cliente_vip('3333333-3', 500, clientes)
True
>>> es_cliente_vip('1234567-8', 150, clientes)
False
```

Programación—Certamen 2 - Jueves 1 de Agosto de 2013

Nombre:

Rol: -

3. [35 %] El director técnico de la selección chilena le entrega como información una **lista de tuplas** con sus jugadores. Cada jugador viene representado por una tupla con la siguiente estructura: (nombre, informacion). A su vez, esta información es una tupla con la siguiente estructura: (posicion, goles, precio). Considere la siguiente estructura **como ejemplo**:

```
jugadores = [  
    #(nombre, (posicion, goles, precio))  
    ('Alexis Sanchez', ('delantero', 8, 14000000)),  
    ('Arturo Vidal', ('central', 3, 4000000)),  
    ('Claudio Bravo', ('arquero', 1, 300000)),  
    ('Gary Medel', ('defensa', 1, 6000000)),  
    ('Gonzalo Jara', ('defensa', 1, 3000000)),  
    ('Eugenio Mena', ('defensa', 0, 5000000)),  
    ('Miguel Pinto', ('arquero', 1, 2000000)),  
    ('Fabian Orellana', ('delantero', 3, 4000000))  
]
```

- a) Implemente la función `jugadores_por_posicion(posicion, jugadores)`, la cual recibe como parámetro un string con la posición y la lista de tuplas `jugadores`. Esta función debe retornar un conjunto con los nombres de los jugadores que se desempeñan en dicha posición.

```
>>> jugadores_por_posicion('delantero', jugadores)  
{'Alexis Sanchez', 'Fabian Orellana'}  
>>> jugadores_por_posicion('defensa', jugadores)  
{'Gary Medel', 'Gonzalo Jara'}  
>>> jugadores_por_posicion('arquero', jugadores)  
{'Claudio Bravo', 'Miguel Pinto'}
```

- b) Implemente la función `cantidad_goles(nombre, jugadores)`, la cual recibe como parámetro un string con el nombre de un jugador y la lista de tuplas `jugadores`. Esta función debe retornar la cantidad de goles realizados por dicho jugador.

```
>>> cantidad_goles('Alexis Sanchez', jugadores)  
8  
>>> cantidad_goles('Eugenio Mena', jugadores)  
0
```

- c) Implemente la función `cantidad_goles_por_posicion(posiciones, jugadores)`, la cual recibe como parámetro un conjunto con las posiciones y la lista de tuplas `jugadores`. Este conjunto tiene la siguiente estructura:

```
posiciones = {'arquero', 'defensa', 'central', 'delantero'}
```

La función retorna un diccionario cuyas claves son las posiciones de los jugadores y su valor la cantidad total de goles realizados por jugadores que se desempeñan en dicha posición.

```
>>> posiciones = {'arquero', 'defensa', 'central', 'delantero'}  
>>> cantidad_goles_por_posicion(posiciones, jugadores)  
{'defensa': 2, 'arquero': 2, 'central': 3, 'delantero': 11}
```