Programación—Certamen 2 - Jueves 30 de Octubre de 2014

Nombre: _____ Rol: _____-_

1. [25 %] Realice el ruteo de los siguientes programas e indique qué es lo que imprimen. Cada vez que el valor de una variable cambie, escríbalo en una nueva fila de la tabla. Recuerde que si una variable es de tipo string, debe colocar su valor entre comillas simples ' '. Si una variable almacena una función, coloque el nombre de ésta como valor (sin comillas).

Importante: La tabla tiene suficientes filas.

```
def f1(x, y=1):
    if y % 2 == 0:
        return x[:y]
    else:
        return x[1:]
def f2(a, b,c):
    m=b[1]
    while m > 1:
        if len(c)>a[0]:
            b = f1(b,m)
            m-=1
        else:
            b = f1(b)
    return b
a,b = (1,2) ,[5,4,3,2,1]
print f2(a,b,f1(b))
```

[4,3]

tuple ((int))

global		f1		f2			
a	b	x	y	a	b	c	m
(1,2)							
	[5,4,3,2,1]						
		[5,4,3,2,1]					
			1				
				(1,2)			
					[5,4,3,2,1]		
						[4,3,2,1]	
							4
		[5,4,3,2,1]					
			4				
					[5,4,3,2]		
							3
		[5,4,3,2]					
			3				
					[4,3,2]		
							2
		[4,3,2]					
			2				
					[4,3]		
							1

Programación—Certamen 2 - Jueves 30 de Octubre de 2014

Nombre: _____ Rol: _____-_

2. [35 %] El mundo está sumido en el caos. Se desató una epidemia, la cual ha arrasado con la mayor parte de la población. Los humanos infectados vuelven a la vida como "walkers", muertos vivientes hambrientos de carne humana. Sin embargo, estos walkers no son el único peligro, sino que muchas veces los humanos deben matar a otros para sobrevivir. En este mundo post-apocalíptico es su misión desarrollar una aplicación en Python para clasificar a los miembros de su grupo de sobrevivientes. Suponga que las estadísticas de grupo se encuentran en el siguiente `diccionario`, cuya clave es el nombre del miembro del grupo y valor una `tupla` con la cantidad de walkers y la cantidad de humanos eliminados (ej: rick ha matado 172 walkers y 10 humanos):

```
grupo = {
        'rick':(172,10), 'daryl':(136,11), 'michonne':(80,6),
        'glenn':(73,0), 'maggie':(55,4), 'carl':(62,1),
        'tyreese':(35,0), 'carol':(17,3) }
```

Obviamente este diccionario es solo un ejemplo, ya que probablemente más de algún miembro morirá mientras usted desarrolla esta aplicación.

a) Desarrolle la función `total(grupo)` que recibe como parámetro un `diccionario` con las estadísticas del grupo, retornando como una `tupla` el total de walkers y el total de humanos eliminados.

```
>>> total(grupo)
(630.0, 35.0)
```

b) Desarrolle la función `puntaje(grupo)` que recibe como parámetro un `diccionario` con las estadísticas del grupo, y retorna un `diccionario` cuya clave es el nombre del miembro del grupo y valor el puntaje asociado. Este puntaje se calcula como: $walkers/total\_walkers + 2 * (humanos/total\_humanos)$, en donde *walkers* representa a los walkers eliminados por el miembro del grupo, y el *total_walkers* representa al total de walkers eliminado por todo el grupo. *humanos* y *total_humanos* sigue la misma lógica.

```
>>> puntaje(grupo)
{'maggie': 0.32, 'glenn': 0.12, 'michonne': 0.47, 'rick': 0.84,
'carl': 0.16, 'carol': 0.2, 'daryl': 0.84, 'tyreese': 0.06}
```

c) Desarrolle la función `ranking(grupo)` que recibe como parámetro un `diccionario` con las estadísticas del grupo, y retorna una `lista` con los nombres de los miembros del grupo ordenados de **mayor a menor** (según el puntaje definido en la parte b).

```
>>> ranking(grupo)
['rick','daryl','michonne','maggie','carol','carl','glenn','tyreese']
```

Programación—Certamen 2 - Jueves 30 de Octubre de 2014

Nombre: _____ Rol: _____-_

3. [40 %] El Aeropuerto-Internacional SansanoAirport desea mantener la información acerca de las salidas y llegadas de sus distintos vuelos; para ello, ha implementado un sistema de información basado en Python, utilizado, desarrollado y mantenido por Sansanos. La base de dicho sistema es un `diccionario` llamado SansaVuelos el cual almacena la información de cada uno de los vuelos.

La clave del `diccionario` es un string que representa el código del vuelo y el valor del diccionario es una `tupla` compuesta respectivamente por el **nombre** de la aerolínea, el lugar de **partida**, **destino**, una `lista` de strings con las **escalas** y el *tiempo_promedio* de todos los vuelos realizados entre los destinos (medido en minutos).

```
SansaVuelos = { # codigo: (nombre, partida, destino, escalas,
   tiempo_promedio),
    'NH217': ('All Nippon Airways', 'Tokyo', 'Santiago', ['Atlanta'],
       1620),
    'AY154': ('Finnair','Helsinki', 'Moscu', ['Riga'], 175),
    'OV171': ('Estonian Air', 'Tallin', 'Paris', ['Amsterdam','Berlin'
       ], 205), ...
}
```

También se cuenta con un `diccionario` Vuelo, cuya clave es un entero **identificador**, y cuyo valor es una `Tupla` compuesta respectivamente por el **código** del vuelo y una **fecha** (en forma de tupla (año,mes,día)).

```
Vuelo = {     # identificador: codigo, fecha
    3542: ('AY154', (2013,12,25)),
    6433: ('NH217', (2013,12,31)),
    1313: ('NH217', (2013,11,6)), ...
}
```

Realice las siguientes funciones para ayudar al correcto funcionamiento de tan respetado aeropuerto:

a) Escriba la función `vuelos_entre_fechas(fecha1,fecha2,Vuelo)` que reciba un par de fechas y el Vuelo y retorne el `conjunto` de todos los códigos de los vuelos que se hayan realizado entre tales fechas **(ambas inclusive)**.

```
>>> vuelo_entre_fechas((2013,7,22),(2014,7,22),Vuelo)
set(['AY154', 'NH217'])
```

b) Escriba la función `vuelos_agotadores(fecha1,fecha2,Vuelos, SansaVuelos)` que reciba un par de fechas y los `diccionarios` Vuelo y SansaVuelos y retorne un `conjunto` con los códigos de los vuelos realizados entre dichas fechas (ambas inclusive) y que sean considerados como agotadores. Un vuelo es considerado agotador si tiene **tres o más escalas** o si su *tiempo_promedio* de vuelo es **superior o igual a 8 horas**.

```
>>> vuelos_agotadores((2013,7,22),(2014,7,22),Vuelo,SansaVuelos)
set(['NH217'])
```

c) Escriba la función `vuelos(partida,destino, Vuelo, SansaVuelos)` que recibe un par de `Strings` con el nombre de la ciudad de partida y la ciudad de destino y los `diccionarios` Vuelo y SansaVuelos. Esta función debe retornar una `lista de tuplas`, en donde cada tupla está compuesta por el código del vuelo, el nombre de la aerolínea y una lista con las fechas en las que se ha efectuado dicho viaje.

```
>>> vuelos('Helsinki','Moscu', Vuelo, SansaVuelos)
[('AY154', 'Finnair', [(2013, 12, 25)])]
```