Programación—Certamen 2 (CSSJ-CC) - Miércoles 31 de Mayo de 2017

Nombre ______________________

Rol ______________ — ___ Paralelo ______

1. [20 %] Realice el ruteo del siguiente programa e indique qué es lo que imprime. 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 ' '.

Importante: *La tabla tiene suficientes filas.*

```
def f1(lista):
  d = {}
  i = 3
  while i > 0:
    for x in lista:
      if x[1] == i:
        if x[1] not in d:
          d[x[1]] = []
        d[x[1]].append(x)
    i -= 1
  return d
```

```
lista=[(1,3),(1,2),(2,1),(2,3)]
print f1(lista)
```

global	f1			
lista	lista	i	x	d

Programación—Certamen 2 (CSSJ-CC) - Miércoles 31 de Mayo de 2017

Nombre

Rol — Paralelo

2. **[40 %]** Inversiones RCaray está siendo investigada por una presunta estafa millonaria. Su presidente generó una asociación ilícita utilizando los dineros de cientos de personas durante los tres últimos años por medio de varias empresas fantasma. Las investigaciones pudieron reunir una información parcial acerca de las personas y las sumas de dineros involucradas. La policía ha recaudado esta información en una lista llamada *estafados*, la que almacena tuplas de la forma $(rut, deuda, empresa, fecha)$, donde rut está en formato string, deuda en formato entero, empresa en formato string y fecha es una tupla (dd, mm, aaaa). A continuación se muestra un ejemplo de la estructura *estafados*:

```
estafados = [
('12.234.567-8', 2000000, 'pelados_furiosos', (25, 5, 2015)),
('9.111.567-k', 5500000, 'multibank', (1, 10, 2014)),
('14.987.007-1', 100000000, 'inversiones_seguras', (30, 11, 2016)),
('12.234.567-8', 10000000, 'multibank', (2, 7, 2015)),
('12.234.567-8', 2500000, 'multibank', (18, 1, 2016)), ...]
```

Una misma persona pudo ser estafada por varias empresas en distintas fechas. De acuerdo a la información anterior, se le solicita implementar las siguientes funciones:

a.- $estafado\_por(lista, rut)$, función que recibe 2 parámetros, la lista de tuplas **estafados** y un string correspondiente al rut de la persona. Esta debe retornar una lista con las distintas empresas que estafaron a la persona con dicho rut. En caso de que el rut ingresado no se encuentre, entregar el mensaje: ”Rut no estafado”.

```
>>> estafado_por(estafados, '12.234.567-8')
>>> ['pelados_furiosos', 'multibank']
>>> estafado_por(estafados, '19.678.222-2')
>>> Rut no estafado
```

b.- $ranking(estafados)$, función que recibe como parámetro la lista de estafados y retorna un diccionario donde cada llave corresponde al nombre de una empresa, y el valor asociado a cada llave es la cantidad de dinero que cada empresa logró acumular.

```
>>> ranking(estafados)
>>> {'inversiones_seguras': 100000000, 'multibank': 18000000,
      'pelados_furiosos': 2000000, ...}
```

c.- $estafados\_por\_fecha(estafados, inicial, final)$, función que recibe como parámetros la lista de estafados, la tupla fecha inicial y la tupla fecha final en formato tuplas (dd, mm, aaaa) y retorna una lista de tuplas (rut, deuda) ordenadas en forma descendente según la deuda total generada entre las fechas entregadas.

```
>>> estafados_por_fecha(estafados,(03,01,2016),(01,01,2017))
>>> [('14.987.007-1', 100000000), ('12.234.567-8', 2500000), ...]
```

NOTA: Los puntos suspensivos en las estruturas de datos indican que existen más datos y que solo se muestran unos pocos a modo de ejemplo.

Programación—Certamen 2 (CSSJ-CC) - Miércoles 31 de Mayo de 2017

Nombre

Rol — Paralelo

3. **[40 %]** Las *Bibliotecas USM* poseen un sistema de préstamo y devolución de libros para alumnos, funcionarios y profesores. Su catálogo de libros está dado por la lista de tuplas `catálogo`, donde cada tupla tiene el número de clasificación, nombre, campus y tipo de préstamo para un libro en particular.

```
catalogo = [('010.245C1', 'Programacion en Python', 'CCV', 'AD'),
            ('145.261D7', 'Algebra Lineal', 'CSJ', 'CB'),
            ('121.748F2', 'Fisica Universitaria', 'CSV', 'CG'), ... ]
```

El tipo de préstamo puede ser `'AD'` (*Alta Demanda*), `'CG'` (*Colección General*) o `'CB'` (*Colección Básica*) con devolución en 2, 7 y 14 días después de la fecha de préstamo, respectivamente. Los préstamos vigentes de libros se manejan en la estructura `prestamos`, donde se guarda el rut del usuario, el número de clasificación y la fecha de préstamo en el formato `(dia, mes, año)`.

```
prestamos = {('12422532-2', '145.261D7', (24, 5, 2017)),
             ('14025935-K', '145.261D7', (13, 5, 2017)),
             ('12422532-2', '010.245C1', (10, 5, 2017)), ... }
```

A continuación, se desea conocer para **una fecha de consulta posterior** a la fecha de todos los préstamos del sistema, la multa que le corresponde a cada usuario por no devolver uno o varios libros a tiempo. Para ello debe crear la función `obtener_multa(catalogo, prestamos, fecha_consulta)` que reciba como parámetros el catálogo de libros, los préstamos registrados en el sistema y la fecha de consulta como una tupla en el formato `(año, mes, dia)`. Esta función debe entregar un diccionario, cuyas llaves corresponden a los ruts de los usuarios de la biblioteca y el valor corresponde a un entero con el monto adeudado. Además, considere lo siguiente:

- La multa por día de atraso corresponde a $1000, $500 y $100 para los libros de Alta Demanda, Colección General y Colección Básica, respectivamente.
- Todos los meses del año tienen 30 días.

Usted puede crear las funciones que estime necesarias para la resolucion de este problema.

```
print obtener_multa(catalogo, prestamos,(2017,6,1))
{'14025935-K': 400, '12422532-2': 19000}
print obtener_multa(catalogo, prestamos,(2017,5,26))
{'12422532-2': 14000}
```