

Programación—Certamen Recuperativo - Lunes 13 de Enero de 2014

Nombre:

Rol: -

1. [25 %] Indique qué es lo que imprimen los siguientes programas.

```
a = 'abracadabra'
b = a.split('a')
b.sort()
print len('\n\n'.join(b))
```

```
d = {'1': 45, '0': 65, '2': '10',
      1: 10, 2:80}
a = 1%3
print d[str(a)] + d[a]
```

```
def f1(x=2):
    return 2**x

def f2(y,z):
    if y%2 == 0:
        return z()
    else:
        return z(y)
```

```
t = 'pepito paga doble\n'
s = t[2::3]
print 'e' in s and not (len(s) %2!=0)
```

```
n = 0
for i in range(4,-1,-1):
    n += f2(i,f1)
print n
```

Pregunta de archivo

1. Considere el archivo `datos.txt`. Indique en el cuadro de más abajo cómo queda el archivo `final.txt` al ejecutar el siguiente segmento de código:

```
def f(x,y,z):
    if z == '-':
        return (x+y)**x
    if z == '+':
        return (x-y)**y
```

```
a = open('datos.txt')
b = open('final.txt', 'w')
for linea in a:
    z,y,x = linea.strip().split(';')
    n = f(int(z),int(y),x)
    t = '{1}{0}\n'
    b.write(t.format(str(n),x))
a.close()
b.close()
```

`datos.txt`

```
2;6;-
2;3;-
10;1;+
3;0;+
```

`final.txt`

Programación—Certamen Recuperativo - Lunes 13 de Enero de 2014

Nombre: Rol: -

2. [35 %] El centro de rehabilitación para personas con necesidades educativas especiales PythonMed, necesita de su ayuda para implementar nuevas funciones relacionadas al módulo nutricional que serán incorporadas al sistema de información que ya existe. Este sistema mantiene los datos de los pacientes (hombres y mujeres) en una estructura como la siguiente:

```
pacientes = {
# rut: (genero, (perimetro cintura, perimetro cadera),
#      {diagnosticos medicos})
  '7839403': ('M', (103.0, 93.0), {'Amaurosis bilateral',
                                'Retardo mental profundo'}),
  '9876543': ('H', (79.0, 84.0), {'Retardo mental profundo', 'Epilepsia'}),
  '3234563': ('M', (62.0, 99.5), {'Epilepsia', 'Tetraparesia espastica'}),
# ...
}
```

Por supuesto, el diccionario puede tener más registros de los que se muestra en el ejemplo. En base a lo expuesto anteriormente, se solicita a usted realizar las siguientes funciones:

- a) Cree la función `diagnostico_frecuente(pacientes)` que reciba el diccionario de pacientes y retorne una lista de tuplas con los diagnósticos médicos y la cantidad de pacientes que los padecen. La lista debe estar en orden descendente en cuanto al número de pacientes que padezcan la enfermedad.

```
>>> diagnostico_frecuente(pacientes)
[('Retraso mental profundo',20), ('Epilepsia',15),
 ('Amaurosis bilateral',2), ...]
```

- b) Cree la función `diagnosticos_comunes(pacientes, n)` que reciba el diccionario `pacientes` y un entero `n`. Esta función debe retornar una lista con el nombre de los `n` primeros diagnósticos que más pacientes los padecen. Si dos o más diagnósticos tienen la misma cantidad de pacientes, seleccione cualquiera.

```
>>> diagnosticos_comunes(pacientes,2)
['Retraso mental profundo', 'Epilepsia']
```

- c) Cree la función `clasificar_pacientes(pacientes)` que reciba el diccionario con los datos de los pacientes y clasifique a los pacientes según la distribución de su tejido adiposo retornando una lista de tuplas. Cada tupla contiene la clasificación y un conjunto con los pacientes dentro de esa clasificación.

Para clasificar a un paciente según la distribución del tejido adiposo se debe calcular el índice cintura cadera (ICC) que está determinado por:

$$ICC = \text{perimetro cintura}(cm) / \text{perimetro cadera}(cm)$$

Según el ICC calculado y la siguiente tabla se puede obtener la clasificación correspondiente.

	Ginecoide	Mixta	Androide
Hombres	< 0,85	0,85 – 1	> 1
Mujeres	< 0,75	0,75 – 0,9	> 0,9

```
>>> clasificar_pacientes(pacientes)
[('ginecoide', set([7362378, 9874322])), ('androide', set([1534672])),
 ('mixto', set([7654238, 3267438, 4888903]))]
```

Programación—Certamen Recuperativo - Lunes 13 de Enero de 2014

Nombre: Rol: -

3. [40 %] Alumnos de la USM han decidido crear su propio servicio de chat, llamado Whatpy, en este se mantiene un archivo con todos los mensajes de los usuarios con el siguiente formato:

Anho/mes/dia hora:minuto:segundo-usuario1-usuario2-mensaje

El archivo siguiente archivo es un **ejemplo** de lo anterior.

mensajes.chat

```
2013/11/08 21:11-cordprogra-elopez-terminaste el ejercicio?
2013/11/08 21:10-cordprogra-elopez-Tranqui, tienen certamen
2014/01/06 20:30-ngalvez-cordprogra-no cachan naa :/
2013/11/08 20:30-elopez-cordprogra-El ayudante no corrige!!
2013/09/20 20:30-ngalvez-elopez-Tenemos el mejor coord :)
```

Considere que el archivo puede tener más líneas que en el **ejemplo**. En base a lo anterior, se le solicita desarrollar las siguientes funciones:

- a) Implemente la función `contar_mensajes(chat)` que reciba el nombre del archivo con los mensajes y retorne un diccionario cuya clave sea el usuario y cuyo valor sea una tupla con la cantidad de mensajes recibidos y cantidad de mensajes enviados.

```
>>> contar_mensajes('mensajes.chat')
{'cordprogra': (2, 2), 'elopez': (3, 1), 'ngalvez': (0, 2)}
```

- b) Implemente la función `mensajes_a(chat, usuario, fecha)` que reciba el nombre del archivo con los mensajes, el nombre de un usuario y una fecha como una tupla de la forma (anho, mes, dia). La función debe crear un archivo `usuario.chat`, donde 'usuario' es el nombre del usuario indicado. Éste debe contener todos los mensajes enviados al usuario con fecha posterior a fecha, la línea se guarda en el mismo formato del archivo chat. La función no retorna nada.

```
>>> mensajes_a('mensajes.chat', 'elopez', (2013,11,18))
```

elopez.chat

```
2013/11/08 21:11-cordprogra-elopez-terminaste el ejercicio?
2013/11/08 21:10-cordprogra-elopez-Tranqui, tienen certamen
```

- c) Implemente la función `dialogo(chat, usuario1, usuario2)` que reciba el nombre del archivo con los mensajes y el nombre de dos usuarios. La función debe crear un archivo `usuario1-usuario2.chat` con los mensajes entre ambos usuarios ordenados de manera cronológica, identificando quien envió el mensaje. No considere el caso que ambos usuarios manden un mensajes al **mismo** tiempo. La función no retorna nada.

```
>>> dialogo('mensajes.chat', 'cordprogra', 'elopez')
```

cordprogra-elopez.chat

```
elopez: El ayudante no corrige!!
cordprogra: Tranqui, tienen certamen
cordprogra: terminaste el ejercicio?
```