

# Programación—Certamen 3 - Lunes 6 de Enero de 2014

Nombre:

Rol: -

1. [30%] Indique que es lo que imprimen los siguientes programas:

```
d='este es una evaluacion final'
e=map(len,d.split())
print max(e)*len(e)
```

```
r = 'Que rico, se vienen las vacaciones'
r.replace(',','!')
r = len(r.split(',') [0])
print r
```

```
a='lineas 2,5,8,9,23'
x=a.split()[1].split(',')
l=map(int,x)
for i in l:
    if i%2==0:
        print i
```

## Preguntas de archivos

1. Considere el archivo `altura.txt`, indique en el cuadro de más abajo qué es lo que se muestra al ejecutar el siguiente segmento de código:

```
archivo = open('altura.txt')
m = 5000
t = 'VOLCAN'
f= 'el {0} {1} es de {2} mts.\n'
for linea in archivo:
    l=linea.strip().split(':')
    x = l[1].split('-')
    if int(x[2])>m and x[1]==t:
        print f.format(t,l[0],x[2])
archivo.close()
```

altura.txt

```
OSORNO:SUR-VOLCAN-2652
PARINACOTA:NORTE-VOLCAN-6342
SAN RAMON:CENTRO-CERRO-3125
PULGAR:NORTE-CERRO-6233
EL PLOMO:CENTRO-CERRO-5530
MAIPO:CENTRO-VOLCAN-5264
VILLARRICA:SUR-VOLCAN-2847
LICANCABUR:NORTE-VOLCAN-5916
CALBUCO:SUR-VOLCAN-2003
```

2. Indique en el cuadro de más abajo cómo queda el archivo `filtro.txt` después de ejecutar el siguiente segmento de código:

```
arch1 = open('todos.txt')
arch2 = open('filtro.txt', 'w')
f = 'R{0}/{1} accidente {2}\n'
for l in arch1:
    l = l.strip().split(';')
    if l[4][0]=='r':
        arch2.write(
            f.format(l[0],l[2],l[5]))
arch1.close()
arch2.close()
```

filtro.txt

todos.txt

```
1;3;1;dia;rural;atropello
2;3;3;noche;rural;choque
1;3;1;noche;urbano;volcamiento
5;3;3;noche;rural;colision
6;4;2;dia;urbano;choque
4;4;5;dia;urbano;choque
5;5;3;dia;rural;atropello
```

## Programación—Certamen 3 - Lunes 6 de Enero de 2014

Nombre:

Rol: -

2. [35 %] Considere una biblioteca digital (virtual) cuya información de libros se encuentra contenida en 3 archivos:

Cada línea del archivo `libros.dat` posee la siguiente estructura: `id_libro`, título, autor, año publicación y el estante en que se ubica físicamente el libro. Suponga que sólo existe una copia por libro.

`libros.dat`

```
...
301:Don quijote de la Mancha:Miguel de Cervantes Saavedra:1605:A3
...
819:Cien Anhos de Soledad:Gabriel Garcia Marquez:1967:C1
...
1019:Santa Maria de las Flores Negras:Hernan Rivera Letelier:1999:C2
```

Cada línea del archivo `palabras_en_libros.dat` posee la siguiente estructura: `palabra`: y un listado de `id_libro`, en donde aparece la palabra, separado por comas. Suponga que este último listado se encuentra ordenado alfabéticamente.

Finalmente, cada línea del archivo `estado_libros.dat` posee la siguiente estructura: el `id_libro` y su estado ('P' en caso de estar prestado y 'D' en caso de estar disponible para préstamo). Suponga que los identificadores se encuentran ordenados ascendentemente.

`palabras_en_libros.dat`

```
...
amor:301,1000,2112
antojadizo:301,1129,5819,6011
tierras:301,303,1120,1129,4903
...
aureliano:819
coronel:819,1019
...
```

`estado_libros.dat`

```
...
301:P
302:D
...
819:D
...
1019:P
1020:P
...
```

**Importante:** Los archivos son en extremo grandes, por lo tanto no es posible almacenar su contenido completo en estructuras estilo diccionario, lista u otros.

Utilizando los archivos presentados se requiere incorporar 3 funcionalidades al sistema de administración de la biblioteca, las que se resumen en 3 subprogramas:

- a) Construir la función `buscar(frase)`, que reciba como parámetro un string, que puede estar compuesto por varias palabras separadas por espacio. Esta función debe retornar una lista con la información de los libros que contienen **por lo menos una** de las palabras.

```
>>> buscar('Coronel Aureliano')
[('Cien Anhos de Soledad', 'Gabriel Garcia Marquez', '1967', 'C1'),
 ('Santa Maria de las Flores Negras', 'Hernan Rivera Letelier',
 '1999', 'C2')]
```

- b) Construir la función `buscar_disponibles(frase)` que reciba como parámetro un string como el del ejemplo anterior. Esta función debe retornar sólo los libros que se encuentren disponibles.

```
>>> buscar_disponibles('Coronel Aureliano')
[('Cien Anhos de Soledad', 'Gabriel Garcia Marquez', '1967', 'C1')]
```

- c) Construir la función `reservar_libro()` que reciba como parámetro un entero, correspondiente al `id_libro`. Esta función debe actualizar el estado del libro en el archivo `estado_libros.dat` a 'P'.

```
>>> reservar_libro(302)
>>>
```

## Programación—Certamen 3 - Lunes 6 de Enero de 2014

Nombre:

Rol: -

3. [35 %] La MafiaFIFA debe realizar el sorteo para su próxima competición mundial por países. Ellos poseen 4 archivos: `bombo1.txt`, `bombo2.txt`, `bombo3.txt` y `bombo4.txt`, donde se encuentran los 32 participantes con su respectiva confederación.

`bombo1.txt`

```
Brasil:CONMEBOL
Espana:UEFA
Argentina:CONMEBOL
Alemania:UEFA
Suiza:UEFA
Colombia:CONMEBOL
Belgica:UEFA
```

`bombo2.txt`

```
Nigeria:CAF
Chile:CONMEBOL
Ghana:CAF
Argelia:CAF
Ecuador:CONMEBOL
C. de Marfil:CAF
Italia:UEFA
Camerun:CAF
```

`bombo3.txt`

```
EEUU:CONCACAF
Australia:ACF
Costa Rica:CONCACAF
Japon:ACF
Corea del S:ACF
Honduras:CONCACAF
Iran:ACF
Mexico:CONCACAF
```

El archivo `bombo4.txt` tiene la misma estructura que los anteriores.

En base a los archivos anteriores le solicita a usted desarrollar las siguientes funciones:

- a) Desarrollar la función `eleccion_aleatoria(archivo)` la cual recibe como parámetro el nombre del archivo correspondiente a uno de los cuatro bombos. Esta función debe retornar al azar una tupla con un país y su respectiva confederación a la que pertenece. Para seleccionar un país al azar use la función `choice` de la librería `random`. (usted debe generar la tupla a partir del archivo).

```
>>> eleccion_aleatoria('bombo2.txt')
('Chile', 'CONMEBOL')
```

- b) Desarrollar la función `sorteo_grupo(grupo)`, esta función recibe un string con el nombre del grupo a crear y debe retornar una tupla con la siguiente estructura (`grupo`, `lista de paises`). El sorteo debe cumplir con las siguientes reglas: el país elegido desde el bombo 1 no puede ser de la misma confederación que el país elegido del bombo 2, el equipo elegido del bombo 1 debe quedar en la posición 1 de su grupo, el resto estar en una posición aleatoria. (debe haber un equipo de cada bombo).

```
>>> sorteo_grupo('Grupo2')
('Grupo2', ['Espanha', 'Holanda', 'Chile', 'Australia'])
```

- c) Desarrollar la función `sorteo_mundial(archivo)` la cual recibe como parámetro el nombre del archivo a crear. Esta función debe generar un archivo con todos los grupos del campeonato, la función no retorna valores. Debe cumplir con que un equipo puede estar en un sólo grupo.

`grupos.txt`

```
>>> sorteo_mundial('grupos.txt')
```

```
GrupoA:Brasil:Croacia:Mexico:Camerun
GrupoB:Espanha:Holanda:Chile:Australia
...
```