

# Programación—Certamen Recuperativo - Jueves 22 de Agosto de 2013

Nombre:

Rol: -

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

```
l = [('a',1), ('b',2), ('c',3)]
for e in l:
    print e[0]*e[1]
```

```
for i in range(2,5,2):
    for j in range(5,3,-1):
        print j, '/', i, 'es', j/i,
        print 'y sobra', j%i
```

```
p = "x;35;y;65;z;60;w;65"
x = map(int,p.split(';')[1::2])
for i in range(2,len(x)):
    print sum(x[:i])
```

```
p = 'onomatopeya'
c = set(p)
d = set(p.split('o'))
print c & d
```

## Preguntas de archivos

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

```
libro = open('quijote.txt', 'r')
encriptado = open('nuevo.txt', 'w')
for i in libro:
    i=i.replace('A', '4')
    i=i.replace('E', '3')
    i=i.replace('I', '1')
    i=i.replace('O', '0')
    encriptado.write(i)
libro.close()
encriptado.close()
```

quijote.txt

```
EN UN LUGAR DE LA MANCHA,
DE CUYO NOMBRE
NO QUIERO ACORDARME,
NO HA MUCHO TIEMPO QUE VIVIA
UN HIDLAGO DE LOS DE LANZA
EN ASTILLERO, ADARGA ANTIGUA,
ROCIN FLACO Y GALGO CORREDOR
```

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

```
def f(x,y):
    return int((x**2+y**2)**0.5)

a = open('data.txt')
b = open('output.txt', 'w')
for line in a:
    x,y = line.strip().split()
    d = f(int(x),int(y))
    b.write(str(d)+'\n')
a.close()
b.close()
```

data.txt

```
1 2
3 4
5 6
```

## Programación—Certamen Recuperativo - Jueves 22 de Agosto de 2013

Nombre:

Rol: -

2. [35 %] La línea aérea LANX desea premiar a sus pasajeros frecuentes incrementando sus kilómetros Lanx-Pass de acuerdo al kilometraje acumulado a la fecha. Para ello, cuenta con el archivo `pasajeros.txt` con campos separados por `:` con la siguiente estructura: `rut:nombre:kilometraje`, tal como se muestra en el siguiente ejemplo:

```
10043499-7:Javier Diaz:553450
7655511-7:Sebastian Rojas:134987
17273221-4:Elizabeth Bugueno:342021
18653834-4:Benjamin Bueno:2000
```

Considere que el archivo anterior puede tener muchos datos. Lo anterior es sólo un ejemplo.

- a) Desarrolle la función `crear_lista(nombre_archivo)` que reciba como parámetro el nombre del archivo y retorne un lista de tuplas con los datos de cada pasajero. **El campo kilometraje debe transformarse a entero.**

```
>>>crear_lista("pasajeros.txt")
[('10043499-7', 'Javier Diaz', 553450),
 ('7655511-7', 'Sebastian Rojas', 134987),
 ('17273221-4', 'Elizabeth Bugueno', 342021),
 ('18653834-4', 'Benjamin Bueno', 2000)]
```

- b) Desarrolle la función `kilometraje(lista)` que reciba como parámetro la lista creada anteriormente, incremente en ella el kilometraje de cada pasajero de acuerdo a la siguiente tabla y que retorne la lista actualizada:

| Kilometraje Actual | Kilómetros a incrementar |
|--------------------|--------------------------|
| 0 - 15.000         | 3.000                    |
| 15.001 - 50.000    | 6.000                    |
| 50.001 - 150.000   | 10.000                   |
| Más de 150.000     | 40.000                   |

```
>>>kilometraje(lista)
[('10043499-7', 'Javier Diaz', 593450),
 ('7655511-7', 'Sebastian Rojas', 144987),
 ('17273221-4', 'Elizabeth Bugueno', 382021),
 ('18653834-4', 'Benjamin Bueno', 5000)]
```

- c) Desarrolle la función `escribir_archivo(nombre_archivo, lista)` que reciba como parámetros un nombre de archivo de salida y la lista actualizada. La función debe guardar la información contenida en la lista en un archivo con el mismo formato que el archivo original, es decir, `rut:nombre:kilometraje`.

```
>>>escribir_archivo("salida.txt", lista)
```

```
10043499-7:Javier Diaz:593450
7655511-7:Sebastian Rojas:144987
17273221-4:Elizabeth Bugueno:382021
18653834-4:Benjamin Bueno:5000
```

## Programación—Certamen Recuperativo - Jueves 22 de Agosto de 2013

Nombre:

Rol:

3. [35 %] El idioma español es mucho más complejo de lo que parece ya que, a diferencia del inglés, los verbos se deben conjugar para cada persona (yo, tú, el, nosotros, vosotros, ellos). Con el fin de enseñar las conjugaciones de los verbos en español se le solicita construir una aplicación en Python. Para ello cuenta con el archivo `verbos.txt` que, por cada línea, tiene la información de un verbo, separada por `;`. El formato es el siguiente: `verbo_infinitivo;conjugaciones`. A su vez, las conjugaciones vienen separadas por `,`. El siguiente es un ejemplo del archivo `verbos.txt`:

```
ser;soy,eres,es,somos,sois,son
ver;veo,ves,ve,vemos,veis,ven
ir;voy,vas,va,vamos,vais,van
```

Considere que el archivo puede tener muchos datos. Lo anterior es sólo un ejemplo.

Adicionalmente, cuenta con la información de pronombres en el siguiente diccionario, que relaciona cada pronombre con su posición o columna, dentro de cada línea del archivo `verbos.txt`.

```
pronombres = {'yo':0, 'tu':1, 'el':2, 'nosotros':3, 'vosotros':4, 'ellos':5}
```

- a) Escriba la función `leer_archivo_verbos(archivo_verbos)`, la cual recibe como parámetro un string `archivo_verbos` que especifica el nombre del archivo en donde se encuentra la información de los verbos, y que retorne un diccionario cuya clave es el verbo en infinitivo y el valor es una lista con todas las conjugaciones que tiene.

```
>>>leer_archivo_verbos("verbos.txt")
{'ver': ['veo', 'ves', 've', 'vemos', 'veis', 'ven'],
 'ser': ['soy', 'eres', 'es', 'somos', 'sois', 'son'],
 'ir' : ['voy', 'vas', 'va', 'vamos', 'vais', 'van']}
```

- b) Escriba la función `conjugar_frase(frase,pronombres,verbos)`, que recibe un string `frase` que representa una frase sin conjugar, el diccionario con los `pronombres` y el diccionario `verbos` el cual corresponde al diccionario obtenido en la función anterior. Debe retornar la frase conjugada correctamente, según el pronombre dado. Guíese por el ejemplo.

```
>>>conjugar_frase("yo ser",pronombres,verbos)
'yo soy'
>>>conjugar_frase("el ir",pronombres,verbos)
'el va'
```

- c) Desarrolle la función `frases_conjugadas(frases,pronombres,verbos)`, que recibe una lista de frases sin conjugar, el diccionario de `pronombres` y el diccionario `verbos` que se obtiene a partir de la función `leer_archivo_verbos("verbos.txt")`. La función no tiene retorno, pero dentro de ella se debe crear un nuevo archivo, llamado `frases.txt`, en donde cada línea representa cada frase conjugada.

```
>>>frases = ["yo ser", "el ir", "nosotros ver"]
>>>frases_conjugadas(frases, pronombres,verbos)
```

```
yo soy
el va
nosotros vemos
```

La función no debe retornar nada, simplemente realizar la operación anteriormente descrita.