

Programación—Certamen 3 (CC) - Lunes 14 de Septiembre de 2015

Nombre: Rol: -

1. [25%] La web Linkedpy analiza los procesos de postulación de recién titulados a empresas. Para ello tiene el archivo `titulados.txt`, donde cada línea tiene a los titulados en el formato `nombre;rut`, y el archivo `postulaciones.txt`, donde cada línea tiene un rut-titulado, el puesto y la empresa a la que cada titulado postula en el formato `rut#puesto#empresa`.

A partir de estos archivos se desea generar un archivo por empresa, los cuales deben tener los titulados que postularon a algún puesto en la empresa con el formato `rut;nombre;puesto`.

A continuación se presentan las líneas de código que resuelven este problema, pero que están desordenadas. Usted debe ordenarlas e indentarlas (dejar los espacios correspondientes de python) para que ambas funciones estén correctas.

La primera función retorna una lista con todas las empresas en el archivo con postulaciones (recibido como parámetro). Y la segunda función resuelve el problema antes descrito, recibiendo como parámetro el nombre del archivo con titulados y el nombre del archivo con postulaciones.

```
def empresas(post):
    emp.append(e)
    arch_P.close()
    for li in arch_P:
        r, p, e = li.strip().split('#')
        if e not in emp:
            arch_P = open(post)
            emp = list()
    return emp
```

```
arch_E = open(e + '.txt', 'w')
for pos in arch_P:
    arch_T.close()
    def registros(tit, post):
        arch_E.write(li.format(r, n, p))
        emp = empresas(post)
        arch_T = open(tit)
        n, r2 = titu.strip().split(';')
        arch_P.close()
        for e in emp:
            arch_E.close()
            if e2 == e:
                r, p, e2 = pos.strip().split('#')
                if r2 == r:
                    li = '{0};{1};{2}\n'
                    arch_P = open(post)
                    return None
        for titu in arch_T:
```

Programación—Certamen 3 (CC) - Lunes 14 de Septiembre de 2015

Nombre:

Rol: -

2. [35 %]

Andrónico Bank es un banco muy humilde que hasta hace poco usaban solo papel y lápiz para manejar toda la información de sus clientes, también humildes. Como una manera de mejorar sus procesos, Andrónico Bank quiere utilizar un sistema computacional basado en Python. Por eso se traspasa la información de sus clientes a un archivo de texto, indicando su rut, nombre y clase cliente. El archivo `clientes.txt` es un **ejemplo** de lo anterior.

`clientes.txt`

```
9234539-9;Sebastian Davalos;VIP
11231709-k;Choclo Delano;Pendiente
5555555-6;Sebastian Pinera;VIP
9999999-k;Gladis Maryn;RIP
12312312-1;Michel Bachelet;VIP
8888888-8;Companero Yuri;Estandar
7987655-1;Sergio Estandarte;RIP
```

A usted se le pide lo siguiente:

- a) Escriba una función `buscar_clientes(archivo, clase)` que reciba como parámetros el nombre del archivo de clientes y una clase, y retorne un diccionario con los rut de los clientes como llaves y los nombres como valor de todos los clientes pertenecientes a la clase entregada como parámetro.

```
>>> buscar_clientes('clientes.txt', 'Pendiente')
{'11231709-k': 'Choclo Delano'}
```

- b) Escriba una función `dar_credito(archivo, rut)` que reciba como parámetros el nombre del archivo de clientes y el rut de un cliente, y que retorne **True** si éste es VIP o **False** si no lo es. Si no encuentra el cliente la función retorna **False**.

```
>>> dar_credito('clientes.txt', '9999999-k')
False
```

- c) Escriba una función `contar_clientes(archivo)` que reciba como parámetros el nombre del archivo de clientes y que retorne un diccionario con la cantidad de clientes de cada clase en el archivo.

```
>>> contar_clientes('clientes.txt')
{'VIP': 3, 'Pendiente': 1, 'RIP': 2, 'Estandar': 1}
```

Programación—Certamen 3 (CC) - Lunes 14 de Septiembre de 2015

Nombre: Rol: -

3. [40 %] Complementando la pregunta 2, se le solicita:

- a) Escriba la función `nuevo_cliente(archivo, rut, nombre, clase)` que reciba como parámetro el nombre del archivo de clientes y el rut, nombre y clase de un nuevo cliente. La función debe agregar el nuevo cliente al final del archivo. Esta función retorna **None**.

```
>>> nuevo_cliente('clientes.txt', '2121211-2', 'Sergio Lagos', 'VIP')
>>>
```

- b) Escriba la función `actualizar_clase(archivo, rut, clase)` que reciba como parámetro el nombre del archivo de clientes, el rut de un cliente y una nueva clase. La función debe modificar la clase del cliente con el rut indicado, cambiándola por `clase` en el archivo. Esta función retorna **True** si logra hacer el cambio o **False** si no encuentra al cliente con el rut indicado.

```
>>> actualizar_clase('clientes.txt', '9234539-9', 'Estandar')
True
```

- c) Escriba una función `filtrar_clientes(archivo, clase)` que reciba como parámetros el nombre del archivo de clientes y una clase de cliente. La función debe crear un archivo `clientes_[clase].txt` con los rut y los nombres de los clientes pertenecientes a esa clase. Note que el archivo debe ser nombrado según la clase solicitada. Esta función retorna **None**.

```
>>> filtrar_clientes('
    clientes.txt', 'VIP')
>>>
```

clientes_VIP.txt

```
5555555-6;Sebastian Pinera
12312312-1;Michel Bachelet
2121211-2;Sergio Lagos
```