Programación—Certamen 3 - Jueves 12 de Diciembre de 2013

Nombre: ______________________ Rol: __________ – __

1. [25 %] Realice el ruteo del siguiente programa e indique el resultado en el cuadro de texto llamado `login.txt`. 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, coloque su valor entre comillas simples ' '. Cuando una variable esté asociada a un archivo, coloque en esta variable el nombre del archivo con el que está trabajando. ***Importante***: *La tabla tiene suficientes filas.*

intentos.txt

```
pedro#pedro
pedro#carcass
pedro#miclave
```

claves.txt

```
Gerardo:clavesegura
pedro:carcass
juan:miclave
```

```
intentos = open('intentos.txt')
login = open('login.txt','w')
logeado = False
for line in intentos:
    claves = open('claves.txt')
    for clave in claves:
        if clave.strip().split(':')[0] == line.strip().split('#')[0]:
            if clave.strip().split(':')[1] == line.strip().split('#')[1]:
                login.write('Bienvenido ' + line.strip().split('#')[0])
                logeado = True
            elif not logeado:
                login.write('Nombre y password no coinciden\n')
    claves.close()

intentos.close()
login.close()
```

login.txt

intentos	login	logueado	line	claves	Clave

Programación—Certamen 3 - Jueves 12 de Diciembre de 2013

Nombre: _____ Rol: _____ – _____

2. **[35 %]** Una minera de cobre dispone de un archivo llamado `zonas.txt`, el cual posee la siguiente estructura por línea:

`Codigo_zona_explotacion:estado_proceso_explotacion:area_de_la_zona`

Ejemplo: `Z1:agotado:15`

Por otra parte, en cada zona de explotación se abren una o más sub-zonas las cuales pueden tener distinta ley de cobre. El detalle de las sub-zonas se almacena en el archivo `subzonas.txt`, el cual posee la siguiente estructura por línea:

`Codigo_zona_explotacion:codigo_sub_zona:ley_cobre_promedio`

Ejemplo: `z3:sz32:1.3`

Los siguientes archivos son un ejemplo de lo anterior:

zonas.txt

```
z1:agotado:15
z2:en explotacion:7
z3:programado:10
z4:programado:30
z5:agotado:5
z6:programado:2
z7:en explotacion:3
z8:en explotacion:9
```

subzonas.txt

```
z1:sz11:0.4
z1:sz12:1.9
z2:sz21:1.1
z2:sz22:0.5
z3:sz31:1.2
z3:sz32:1.3
z4:sz41:0.5
z4:sz42:0.2
z5:sz51:0.8
z6:sz61:1.1
z6:sz62:1.6
z7:sz71:1.6
z8:sz81:1.1
```

Se le solicita a usted desarrollar las siguientes funciones:

a) a) Codificar la función `zonas(area, archivo)` que retorne una lista de las zonas en explotación con superficie mayor al área dada como parámetro.

```
>>> print zonas(7, 'zonas.txt')
['z1', 'z3','z4','z8']
```

b) Codificar la función `ley_promedio(zonas, archivo)` que retorne la ley promedio de una lista de zonas dada como parámetro.

```
>>> print ley_promedio(['z1','z2'], 'subzonas.txt')
0.975
```

c) Codificar la función `mejor_zona(Estado, archivo1, archivo2)`, la cual recibe como parámetro un String con el estado de la zona y el nombre de los archivos. Esta función debe retornar una tupla con el código de la zona que cumple con el estado dado como parámetro y el mejor promedio de ley de todas las sub zonas.

```
>>> print mejor_zona('Programado','zonas.txt','subzonas.txt')
('z6', 1.35)
```

Programación—Certamen 3 - Jueves 12 de Diciembre de 2013

Nombre: _____ Rol: _____ – _

3. **[40 %]** El servicio nacional del consumidor del país de Pitonia, en su afán de automatizar y fidelizar sus procesos de comparación de precios, acudió a usted para solicitar su ayuda. Actualmente la comparación es manual y desean solicitar a cada organización un archivo de texto con la siguiente estructura.

La primera línea corresponde a la ubicación en el plano del establecimiento (coordenadas x e y). La segunda línea corresponde a la descripción de los productos, comenzando con: `nombreProd1#categoria#stock#precio`.

falaferia.txt

```
44.6#5.8
Ipad#Tecnologia#23#199990
BicicletaTrek#Deporte#150#101500
Netbook#Tecnologia#100#156990
...
```

En base a la estructura del archivo anterior le solicita a usted desarrollar las siguientes funciones:

a) Cree la función `obtener_informacion(archivo,producto)`, la cual recibe como parámetro el nombre de un archivo y el nombre de un producto. Esta función retorna una tupla con la ubicación del establecimiento como tupla y el precio del producto, si el producto no se encuentra en el establecimiento la función debe retornar **`False`**.

```
>>> print obtener_informacion('Falaferia.txt','Ipad')
((44.6,5.8), )
```

b) Cree la función `calcular_distancia(lista, rut, ubicacion, producto)`, la cual recibe como parámetro una lista con los nombres de los establecimientos, un String rut correspondiente al rut del cliente, una tupla ubicación con las coordenadas x e y de la ubicación del cliente y un String con el nombre de un producto. Esta función crea un archivo de texto cuyo nombre es el rut del cliente, (ver ejemplo), con la siguiente estructura:
`nombre_producto@precio@nombre_establecimiento@distancia`. Donde distancia corresponde a la distancia entre el cliente y el establecimiento. Este archivo tendrá tantas líneas como establecimientos tengan el producto que se busca. Si el producto no se encuentra en algún establecimiento no se debe agregar al archivo. La función retorna **`True`** si el producto se encontró en algún establecimiento y **`False`** en caso contrario.

```
>>> calcular_distancia(['Falaferia',
'Sansanito','Python Market'],
'11111111-1',(87.5, 45.7), 'Ipad')
True
```

11111111-1.txt

```
Ipad@299590@Sansanito@7.3
Ipad@199990@Falaferia@15
Ipad@230000@Python Market@4.5
```

c) Cree la función `recomendar_establecimiento(lista,rut,ubicacion, producto)`, la cual recibe como parámetro una lista con los nombres de los establecimientos, un String correspondiente al rut del cliente, la tupla ubicacion con las coordenadas x e y de la ubicación del cliente y un string con el nombre de un producto. Esta función crea el archivo llamado `recomendaciones.txt`, el cual posee la estructura presentada en el ejemplo. La función retorna **`True`**, si encontró el producto en algún establecimiento y pudo crear el archivo y **`False`** en caso contrario.

```
>>> print recomendar_establecimientos(['Falaferia','Sansanito',
'Python Market'], '11111111-1',(87.5, 45.7), 'Ipad')
True
```

recomendaciones.txt

```
Producto: Ipad
Establecimiento con menor precio: Falaferia
Precio: $199990
Se encuentra a: 15 Km de distancia.
Establecimiento mas cercano que tiene el producto: Python Market
Precio: $230000
Se encuentra a: 4.5 km de distancia.
```