

Programación—Certamen 3 (CSJ y CV) - Martes 25 de Agosto de 2015

Nombre: Rol: -

1. [25 %] El Dpto de promoción de la USM necesita identificar los alumnos que participaron en los ensayos PSU del 2014 y que están matriculados este 2015, para lo cual posee dos archivos de texto, `matriculados.txt` y `ensayos.txt`, cada uno con la estructura descrita más abajo

`matriculados.txt`

```
Rut#Nombre#apellido#mail#carrera
```

`ensayos.txt`

```
Rut,nombre,ensayo,aaaa-mm-dd
```

A partir de estos archivos se desea generar uno nuevo, el cual debe tener los alumnos que participaron en algún ensayo PSU y que actualmente se encuentra matriculado en la USM. Los datos que le interesa tener son:

```
Rut#nombre#apellido#mail#carrera#cantidad_ensayos
```

A continuación se presentan las líneas del 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.

Tener en cuenta que la función `num_ensayos` debe retornar la cantidad de ensayos en la que participó la persona (rut) pasado como parámetro.

La función `crear_archivo` recibe 2 parámetros, el nombre del archivo con los alumnos matriculados y el nombre del archivo con los alumnos que participaron en algún ensayo. La función debe crear el archivo `Reporte.txt`, con los alumnos que están matriculados y que participaron en al menos 1 ensayo.

`num_ensayos`

```
arch.close()
def num_ensayos(rut, archivo):
    if rut == l[0]:
        arch = open(archivo)
        l = linea.strip().split(",")
        for linea in arch:
            cont+=1
        return cont
    cont = 0
```

`crear_archivo`

```
new.close()
new = open("Reporte.txt", "a")
def crear_archivo(matri, ensa):
    x = num_ensayos(l[0], ensa)
    for linea in arch:
        arch.close()
        if x>0:
            s = linea.strip()
            new.write(s+","+str(x)+"\n")
        arch = open(matri)
        l = linea.strip().split("#")
```

Programación—Certamen 3 (CSJ y CV) - Martes 25 de Agosto de 2015

Nombre: Rol: -

2. [35 %] Se le solicita a usted realizar un programa que ayude a los padres jóvenes en la selección de colegios para sus hijos, para lo cual usted dispondrá de 2 archivos:

- `resultados.txt` que contiene un código del colegio, religión, arancel, coordenadas x e y, grado de inglés (1 a 5, donde 5 es intensivo), promedio de Simce de 8 básico, promedio PSU del último año.
- `colegios.txt` que contiene el código y los nombres de los colegios.

`colegios.txt`

```
01-Colegio el Salvador
02-Colegio Santa Ines
03-Instituto Regional
04-Python College
05-Manuel of Classroom
06-Pythonsianos
07-Python F14
...
```

`resultados.txt`

```
01@Catolica@99990@145.4#-90.6@3@214@676
07@Laico@54500@-234.9#34.23@1@154@567
04@Evangelica@145990@-32.5#12.1@5@196@625
02@Catolica@87590@-99.3#345.2@2@166@599
06@Catolica@121350@-14.67#76.0@3@201@615
03@Laico@103990@0.9#102.8@4@235@750
05@Evangelica@95980@56.9#-23.1@3@178@603
...
```

En base a los archivos presentados anteriormente, se le solicita:

- a) Desarrolle la función `agregar_colegio(arch1, arch2, cod, nombre, religion, arancel, ubicacion, ingles, simce, psu)`, la cual recibe los siguientes parámetros en formato String: el nombre de los archivos de colegios (`arch1`) y resultados (`arch2`), el código, nombre y religión del colegio que se quiere agregar. Los siguientes parámetros se entregan en formato entero: `arancel`, grado de inglés, promedio `simce` y `PSU`. Además se recibe una tupla de flotantes con la ubicación (`cordx, cordy`).

Esta función no retorna nada, solo agrega a los archivos respectivos el colegio.

```
>>> agregar_colegio("colegios.txt", "resultados.txt", "11", "Python School", "Laico", 79990, (-15.3, 105.2), 4, 175, 621)
```

- b) Desarrolle la función `actualizar_colegio(archivo, codigo, arancel, ingles, simce, psu)`, la cual recibe como parámetro un string con el nombre del archivo, un string con el código del colegio y el nuevo arancel, grado de inglés, promedio `SIMCE` y `PSU` en formato entero para el colegio a actualizar. La función debe retornar `True` indicando que el archivo se actualizó correctamente y `False` en caso de que el colegio no se encuentre en el archivo.

```
>>> actualizar_colegio("resultados.txt", "03", 149990, 5, 245, 758)
True
>>> actualizar_colegio("resultados.txt", "63", 49990, 2, 156, 558)
False
```

Programación—Certamen 3 (CSJ y CV) - Martes 25 de Agosto de 2015

Nombre: Rol: -

3. [40 %] En base a los archivos presentados en la pregunta 2, se le solicita:

- a) Desarrolle la función `clasificar(archivo)`, la cual recibe como parámetro el nombre del archivo de texto. La función debe retornar un diccionario, cuya clave es el nombre de una religión y cuyo valor es un conjunto con los códigos de los colegios que son de dicha religión.

```
>>> clasificar("resultados.txt")
{'Laico': {'07', '03', '11'},
 'Catolica': {'01', '02', '06'},
 'Evangelica': {'04', '05'}}
}
```

- b) Desarrolle la función `filtrar(arch1, arch2, religion, apoderado, umbral)`, la cual recibe como parámetro el nombre de los archivos `arch1` y `arch2`, un string con la religión de interés, una tupla `apoderado` que corresponde a la ubicación en el plano de la casa del apoderado (`cordx, cordy`) y un entero con el valor de la distancia máxima.

La función debe generar un archivo con la información de los colegios que pertenecen a dicha religión y que se encuentren a una distancia menor o igual que al valor umbral dado como parámetro.

El formato del archivo es el siguiente:

`Nombre_colegio#arancel#distancia#PSU`

```
>>> filtrar("resultados.txt", "colegios.txt", "Laico", (10, -15), 25)
```

`filtro.txt`

```
Instituto Regional#103990#19.4#750
Python School#79990#24.6#621
```

HINT: La distancia entre el punto (x_1, y_1) y el punto (x_2, y_2) está definida por:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$