

## Programación – Certamen 3 - Martes 11 de Diciembre de 2018

Nombre

Rol

Paralelo

Voronoi

1. [20%] El problema ambiental en Quintero es de preocupación nacional, razón por la cual se ha estado estudiando de cerca la zona y midiendo en forma constante los niveles de distintos gases nocivos para la salud. Como estudiante sansano, te solicitan ayuda para generar una función que facilite el monitoreo del aire a partir de datos existentes.

En primer lugar, el Ministerio de Salud te provee el archivo de texto `datos.txt` con las mediciones de tres elementos ( $SO_2$ ,  $MP10$  y  $NO_2$ , siempre en ese orden) en distintas industrias de la zona, en determinadas fechas:

`datos.txt`

```
Enab,07/09/2018,S02,4,MP10,0,NO2,5
Oxy,07/09/2018,S02,8,MP10,8,NO2,8
GML,07/09/2018,S02,12,MP10,10,NO2,0
Cudelco,07/09/2018,S02,5,MP10,10,NO2,5
Enab,08/09/2018,S02,0,MP10,5,NO2,8
Oxy,08/09/2018,S02,10,MP10,8,NO2,8
...
```

Desean generar un diccionario con esta información, cuya llave sea la fecha y el valor sea una lista de tuplas. Cada tupla debe tener como primer elemento el nombre de la empresa, y como segundo elemento un diccionario. Este diccionario tendrá como llaves los elementos  $SO_2$ ,  $NO_2$  y  $MP10$  y el valor de cada una corresponderá a la medición respectiva según el archivo de entrada.

Ordena e indenta correctamente las siguientes líneas de código, que permiten crear la estructura de datos solicitada.

```
archivo.close()
datos = linea.strip().split(',')
fecha = datos[1]
gas['MP10'] = int(datos[5])
dat_emp[fecha].append((nom, gas))
dat_emp = {}
def med_fecha():
    gas = {'SO2':int(datos[3])}
    nom = datos[0]
    if fecha not in dat_emp:
        dat_emp[fecha] = []
    for linea in archivo:
        gas['NO2'] = int(datos[7])
    archivo = open('datos.txt')
    return dat_emp
```

**NOTA:** Los puntos suspensivos indican que existen más datos.

## Programación – Certamen 3 - Martes 11 de Diciembre de 2018

Nombre

Rol

--	--	--	--	--	--	--	--	--	--

—

Paralelo

--	--	--

Voronoi

2.

### Programación – Certamen 3 - Martes 11 de Diciembre de 2018

Nombre

Rol

--	--	--	--	--	--	--	--	--	--

—

Paralelo

--	--	--

Voronoi

**3.**

## Programación – Certamen 3 - Martes 11 de Diciembre de 2018

2. [40 %] **NOTA:** Suponga que el archivo de datos no contiene errores y se apega al formato especificado.

a.- **IMPORTANTE:** En este ítem **debe** llamar a la función definida en la pregunta 1.

Ahora que es posible procesar los datos del archivo, el Ministerio de Salud desea obtener más información a partir de ellos.

En primer lugar, se desea saber cuál es la empresa que más contaminó para cierto tipo de gas en una fecha dada. Desarrolle la función `mas_cont (fecha, gas)` para conseguirlo. El primer parámetro corresponde a la fecha que se desea considerar. El segundo parámetro es un string que representa un gas ( $SO_2$ ,  $NO_2$  o  $MP10$ ). La función debe retornar una tupla, cuyo primer elemento es el nombre de la empresa que más contamina para el `gas`, y el segundo elemento es el valor marcado por el `gas` en la `fecha` señalada. En caso de empate, la función debe retornar una tupla con cualquiera de las empresas que más contaminan para la fecha indicada. Puede suponer que el archivo contiene datos para todas las `fechas` que serán consultadas.

Por ejemplo, considerando los datos anteriores:

```
print mas_cont('07/09/2018', 'NO2')
>> ('Oxy', 8)
```

b.- Finalmente, es importante poder relacionar los datos que tienen los hospitales con la información de contaminación. Los centros hospitalarios tienen registros de los contaminantes que han producido más episodios de intoxicación en la población, almacenados como una lista de tuplas (ver ejemplo: `dat_hos`). Cada tupla tiene como primer elemento la fecha del incidente, y como segundo elemento el contaminante ( $SO_2$ ,  $NO_2$  o  $MP10$ ) que lo produjo. Desarrolle la función `estadisticas (dat_hos)` que cree un archivo de texto a partir de esta lista de tuplas. El archivo resultante debe mostrar en cada línea el día, la industria que más emisiones del contaminante tuvo en ese día, el contaminante y su nivel registrado.

Por ejemplo, considerando los datos anteriores:

```
dat_hos = [('07/09/2018', 'NO2'), ('07/09/2018', 'SO2')]
estadisticas(dat_hos)
```

estadisticas.txt

```
07/09/2018 Oxy NO2 8
07/09/2018 GML SO2 12
```

**RECORDATORIO:** Además de lo visto para los certámenes anteriores, puede utilizar, entre otras funciones:

```
Para un string s: s.upper(), s.lower(), s.replace(actual,nuevo), s.strip(),
s.split(separador), s.format(valor1, valor2...), separador.join(lista_strings)
Operaciones de archivo: open(archivo,modo), a.close(), a.write(s)
Sean c1 y c2 conjuntos: c1.add(e), c1.remove(e), c1 & c2 (intersección),
c1 | c2 (unión), c1 - c2 (diferencia), c1 < c2 (es subconjunto propio),
e in c1 (pertenencia), len(c1)
```

## Programación – Certamen 3 - Martes 11 de Diciembre de 2018

3. [40%] Durante el verano usted y sus amigos han decidido pasar algún tiempo realizando actividades en la naturaleza. Para ello, obtuvo el archivo `lugares.txt`, que contiene en cada línea el nombre de algún lugar con sus características naturales (`lugar:naturaleza1,naturaleza2,...`).

`lugares.txt`

```
pochoco:cerro
saltos:sendero,rios,cerro,cascada
aculeo:laguna,rios
el cajon:valle,rios
...
```

`actividades.txt`

```
acampar:el cajon,aculeo,pochoco,saltos
pescar:aculeo,el cajon
escalada:el cajon,pochoco
rafting:el cajon,saltos
...
```

Además, cuenta con el archivo `actividades.txt` que contiene en cada línea el nombre de alguna actividad al aire libre seguido de los lugares donde es posible realizar dicha actividad en el formato: `actividad:lugar1,lugar2,...`

Para tomar la decisión de donde ir se le pide desarrolle la función `recomendar(cj_nat, cj_act)` que recibe el conjunto `cj_nat` con ciertas cosas naturales que debe tener el lugar escogido y el conjunto `cj_act` con ciertas actividades que deben poder realizarse en el lugar escogido. La función debe retornar una **lista con el nombre** de los lugares que incluyan **TODA** la naturaleza del conjunto `cj_nat` y en donde se puedan desarrollar **TODAS** las actividades del conjunto `cj_act`. Si ningún lugar cumple con todos los requisitos, la función debe retornar una **lista vacía**.

Por ejemplo, considerando los datos de los archivos mostrados anteriormente:

```
>>> recomendar({'rios','sendero'}, {'acampar'})
['saltos']
>>> recomendar({'cerro'}, {'pescar'})
[]
>>> recomendar(set(), {'acampar','pescar'})
['el cajon', 'aculeo']
```