

# Programación – Certamen Recuperativo - Viernes 9 de Agosto de 2019

Nombre

Rol

Paralelo

Newell

1. [20%] Un terrible virus informático atacó la única aplicación de mensajería instantánea utilizada en Pythonia. Todos los mensajes enviados a tus contactos se han desordenado y sólo queda un registro similar al siguiente:

mensajes.txt

```
22:30/Pedro/Se mejore
14:54/Federico/A tus comentarios
07:00/Pedro/Que
14:52/Federico/Quedo atento
...
```

Cada línea contiene la hora de envío de un mensaje, el nombre del contacto a quién fue enviado el mensaje, y su contenido. Todos estos campos están separados entre sí por una barra diagonal ("/").

Para tratar de reparar el daño causado por el virus, se cuenta con la función `contactos(archivo)` que recibe un string con el nombre de un archivo como el anterior y retorna un diccionario cuyas llaves sean el nombre de todos los contactos y su valor una lista de tuplas con la hora y contenido de todos los mensajes enviados a ese contacto. Estudie el siguiente ejemplo para comprender lo que hace la función:

```
>>> contactos('mensajes.txt')
{'Pedro': [('22:30', 'Se mejore'), ('07:00', 'Que')],
 'Federico': [('14:54', 'A tus comentarios'), ('14:52', 'Quedo atento')]}
```

## Ordenamiento

A continuación se muestran las instrucciones que forman el código de la función `contactos`, pero se encuentran desordenadas y sin indentación. Usted debe **ordenarlas** e **indentarlas** adecuadamente para que la función lleve a cabo el proceso descrito. Escriba la función completa a la derecha del código desordenado. No debe agregar nuevas instrucciones y todas las instrucciones dadas deben ser utilizadas.

```
def contactos(archivo):
dest = lista[1]
return d
d[dest].append((hora,msg))
d = {}
msg = lista[2]
for line in a:
d[dest] = []
hora = lista[0]
a.close()
if dest not in d:
a = open(archivo)
lista = line.strip().split('/')
```

## Programación – Certamen Recuperativo - Viernes 9 de Agosto de 2019

Nombre

Rol

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

—

Paralelo

--	--	--

Newell

2.

**Programación – Certamen Recuperativo - Viernes 9 de Agosto de 2019**

Nombre

Rol

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

—

Paralelo

--	--	--

Newell

**3.**

## Programación – Certamen Recuperativo - Viernes 9 de Agosto de 2019

2. **NOTA:** Puede utilizar la función `contactos` de la Pregunta 1. No es necesario que la copie de nuevo.

[35%] Siguiendo el contexto anterior, escriba la función `ordenar(archivo)` que recibe un *string* con el nombre de un archivo como el descrito anteriormente. Esta función debe crear un archivo de nombre `chat.txt` cuyo contenido tendrá la siguiente estructura, para cada contacto: el nombre de un contacto en mayúsculas y en la línea siguiente los mensajes enviados a ese contacto **ordenados** por hora de envío. Los datos de distintos contactos deben estar separados entre sí por una línea en blanco. Además, la función debe retornar el número total de contactos procesados. Guíese por el ejemplo a continuación.

Ejemplo:

```
>>> print(ordenar('mensajes.txt'))  
2
```

chat.txt

```
PEDRO  
Que Se mejore  
  
FEDERICO  
Quedo atento A tus comentarios
```

## Programación – Certamen Recuperativo - Viernes 9 de Agosto de 2019

3. [45%] Desde el Aeropuerto Internacional de Pythonia salen constantemente vuelos a las principales ciudades del mundo. La información de los vuelos que saldrán en un día en particular se encuentran almacenados en el archivo `salidas-a-m-d.csv`, cuyo nombre incluye el año, mes y día correspondiente. Las líneas del archivo tienen el formato: **vuelo;destino;hora**. El código del vuelo está formado por 2 letras que identifican de manera única a la línea aérea, seguidas de una serie de dígitos. Por ejemplo:

`salidas-2019-08-09.csv`

```
UA7998;Tokyo;12:00
AA6991;Toronto;12:05
VS4667;Chicago;12:20
AA291;Toronto;12:59
AA3492;Chicago;12:20
UA23;Toronto;16:25
AA6250;Madrid;17:05
BA2492;Chicago;17:20
...
```

Escriba la función `agrupar( fecha )`, que recibe como parámetro una tupla de *strings* con formato (aaaa, mm, dd). La función debe crear un archivo llamado `destinos.txt`, que contenga una línea para cada destino distinto encontrado en el archivo de entrada, ordenado de mayor a menor según la cantidad de aerolíneas que vuelan a ese destino.

El formato de cada línea del archivo `destinos.txt` es: **destino:num**, donde **num** corresponde a la cantidad de aerolíneas distintas que vuelan a esa ciudad.

Además de crear el archivo descrito, la función debe retornar el número de destinos distintos que fueron identificados.

Guíese por el ejemplo. En particular, estudie lo que pasa con `Toronto`. A pesar de que hay 3 vuelos a ese destino sólo hay 2 aerolíneas distintas que vuelan ahí, por lo que el archivo de salida indica 2.

Ejemplo:

```
>>> print(agrupar(('2019', '08', '09')))
4
```

`destinos.txt`

```
Chicago:3
Toronto:2
Tokyo:1
Madrid:1
```