

Programación – Certamen Recuperativo - Jueves 7 de Enero de 2021

Instrucciones

Este certamen está compuesto por 2 preguntas de desarrollo y 1 video.

Usted dispondrá de 150 minutos para completar las preguntas de desarrollo y enviarlas a través de los buzones habilitados en Aula, esto es, hasta las 19:50. El video podrá ser entregado hasta las 21:00 horas, subiendo el enlace respectivo en un buzón de Aula habilitado para ese efecto.

Suba su video a *YouTube* o a *Google Drive*. Si utiliza *YouTube*, asegúrese de que quede **Oculto**, pero por ningún motivo público o privado. Si utiliza *Google Drive*, asegúrese de dejarlo visible para quien tenga el *link*. No publique su video por ningún otro medio o red social.

Entrega

Estarán disponibles 3 buzones de entrega: uno para cada una de las preguntas y uno para que entregue el enlace al video.

En el caso de las preguntas 1 y 2, debe subir el archivo con el código de su solución, es decir, con **extensión .py**. No suba archivos comprimidos (.zip). En caso de utilizar *repl.it*, que descarga un archivo .zip, asegúrese de descomprimirlo y enviar únicamente el archivo .py.

Instrucciones relacionadas al video

Grabe **un único video**, cuya duración debe ser de entre 3 y 5 minutos, explicando la forma en que resolvió únicamente la **pregunta 2**.

NO debe grabarse escribiendo el código, sino que una vez que lo finalice, comente las estrategias que utilizó para resolver el problema. Apoye sus explicaciones señalando las partes del código correspondientes.

Para orientarle en la grabación del video, le recomendamos guiarse con las siguientes preguntas:

- ¿Cómo clasificó las marcas de acuerdo país de origen?
- ¿Cómo filtró sólo aquellas marcas que cumplen con el rubro y año indicados?
- ¿Cómo consiguió quedarse únicamente con el mayor valor de cada marca en el año?
- ¿Cómo consiguió ordenar la información dentro de cada archivo de salida?

Ponderación del video

Es importante que grabe y envíe el video incluso si la pregunta 2 está parcialmente completa. Esto, porque el video es un ponderador sobre la nota de esta pregunta:

Sea nd la nota obtenida en el desarrollo de la pregunta 2, antes de la revisión del video:

- Si no adjunta video, $nota_pregunta = nd \times 0 = 0$
- Si adjunta el video, y
 - * Explica correctamente su código en la grabación y demuestra comprensión de diccionarios, listas, procesamiento de texto y archivos, $nota_pregunta = nd \times 1 = nd$
 - * Hay un esfuerzo por darse a entender en la grabación, y evidencia comprensión de diccionarios, listas, procesamiento de texto y archivos, $nota_pregunta = nd \times 0,7$
 - * La explicación grabada consiste en una lectura del código, sin profundizar en su razonamiento, $nota_pregunta = nd \times 0,3$

Programación – Certamen Recuperativo - Jueves 7 de Enero de 2021

1. [30 %] Se cuenta con un archivo que contiene la población de los distintos países del mundo, registrada en distintos años a lo largo de la historia. Las líneas del archivo tienen el siguiente formato:

año, ciudad, país, población, región

La población está dada en miles de habitantes. A modo de ejemplo, a continuación se muestra una parte del archivo descrito:

ciudades.txt

```
1977,Moscow,Russia,7828,Europe
1973,Chicago,United States,7594,North America
1979,New York,United States,15901,North America
1834,New York,United States,341,North America
2018,New York,United States,18713,North America
1882,New York,United States,2556,North America
1883,Chicago,United States,825,North America
1752,Hangzhou,China,342,East Asia
...
```

Observe que las ciudades pueden aparecer varias veces, en distintos años. Asimismo, los países tienen una cantidad diferente de ciudades.

Escriba la función `min_max(nombre_archivo, pais)`, que recibe como parámetro un *string* con el nombre del archivo que contiene los datos de las ciudades, y un *string* con el nombre de un país particular. La función debe retornar una lista de tuplas, donde cada tupla corresponde a una ciudad del país y debe tener la siguiente estructura:

`(ciudad, año_menor_pob, menor_población, año_mayor_pob, mayor_población)`

Como sugieren los nombres, cada tupla contiene el nombre de la ciudad, el año en que se contabilizó la menor población para esa ciudad, la menor población, el año en que se registró la mayor población, y la mayor población. Puede suponer que nunca habrá empates.

Las tuplas pueden aparecer en la lista, en cualquier orden.

Ejemplo:

```
>>> print(min_max('ciudades.txt', 'United States'))
[('Philadelphia', 1851, 441, 1937, 2586), ('New York', 1834, 341, 2018, 18713),
('Chicago', 1883, 825, 1973, 7594), ('Los Angeles', 1938, 2654, 2003, 11906),
('Boston', 1924, 1722, 1928, 1858)]
```

Programación – Certamen Recuperativo - Jueves 7 de Enero de 2021

2. [70 %] El valor de marca es un indicador monetario que mide la fuerza de una marca en el mercado, a partir de los activos financieros que la rodean. Tenemos un archivo que registra el valor de marca de muchas empresas a nivel mundial, en distintas fechas. Las líneas de este archivo tienen el siguiente formato:

año-mes-día,marca,país,valor,rubro

Observe que los elementos que componen cada fecha están separados por guiones. Además de la marca y su valor, cada línea registra también el país de origen de la marca y el rubro al que pertenece, que es un texto en Inglés.

A modo de ejemplo, a continuación se muestra una parte del archivo descrito:

marcas.txt

```
2016-01-01,Kellogg's,United States,11711,Fast-Moving Consumer Goods
2016-05-01,Kellogg's,United States,11465,Fast-Moving Consumer Goods
2017-01-01,LEGO,Denmark,7024,Fast-Moving Consumer Goods
2007-03-01,Nokia,Finland,34070,Technology
2016-02-01,LEGO,Denmark,6719,Fast-Moving Consumer Goods
2012-12-01,Zara,Spain,10710,Apparel
2016-06-01,LEGO,Denmark,6830,Fast-Moving Consumer Goods
2016-12-01,LEGO,Denmark,6996,Fast-Moving Consumer Goods
2016-02-01,Hyundai,South Korea,12601,Automotive
...
```

Escriba la función **separar(nombre_archivo, rubro, año)**, que recibe como parámetro un *string* con el nombre del archivo que contiene los datos de las marcas, un *string* con un rubro particular y un número entero que representa un año. La función debe considerar únicamente marcas del rubro y año que fueron indicados en los parámetros, y debe crear un archivo para cada país que contenga marcas que cumplen el filtro.

Los archivos deben contener las marcas ordenadas de mayor a menor valor, considerando cada marca una única vez, con el mayor valor. El nombre de cada archivo debe ser igual al nombre del país, seguido del sufijo `.txt`. Por ejemplo, `Japan.txt`.

El formato de las líneas de los archivos debe ser idéntico al que se muestra en el ejemplo a continuación. En particular, observe que el valor de las marcas aparece en dólares y no en miles como se encontraba en el archivo original.

Además de crear los archivos señalados, la función debe retornar el número total de países distintos que se consideraron, es decir, la cantidad de archivos que fueron creados.

Puede suponer que el rubro indicado en el parámetro siempre existe y que se encontrará al menos una marca de ese rubro para el año del parámetro.

A continuación se muestra el comportamiento que debe tener la función, y el contenido de los archivos resultantes.

Ejemplo:

```
>>> print(separar('marcas.txt', 'Fast-Moving Consumer Goods', '2016'))
4
```

United States.txt

```
Mayor valor en 2016 de las marcas del rubro Fast-Moving Consumer Goods:
Marca: Gillette, valor en US$: 19950000
Marca: Pampers, valor en US$: 16392000
Marca: Kellogg's, valor en US$: 11711000
Marca: Colgate, valor en US$: 8413000
Marca: Johnson & Johnson, valor en US$: 6020000
```

Programación – Certamen Recuperativo - Jueves 7 de Enero de 2021

Denmark.txt

Mayor valor en 2016 de las marcas del rubro Fast-Moving Consumer Goods:
Marca: LEGO, valor en US\$: 6996000

Switzerland.txt

Mayor valor en 2016 de las marcas del rubro Fast-Moving Consumer Goods:
Marca: Nestlé, valor en US\$: 8726000

France.txt

Mayor valor en 2016 de las marcas del rubro Fast-Moving Consumer Goods:
Marca: L'Oréal, valor en US\$: 10930000
Marca: Danone, valor en US\$: 9312000