

Programación – Cert. Recuperativo - Martes 18 de Diciembre de 2018

Nombre

Rol

—

Paralelo

SantaClaus

2.

Programación – Cert. Recuperativo - Martes 18 de Diciembre de 2018

2. [40%] NOTA: Puede hacer uso de la función definida en la pregunta 1.

Queremos contar con un programa que ayude a los apoderados a comprar los libros de la lista de útiles de sus hijos, pagando el menor precio posible por ellos. Para ello se consultará los precios en distintas librerías eligiendo siempre comprar en la que vende cada libro al menor precio.

El inventario de las librerías se guarda en un diccionario cuya llave es el nombre de la librería. Asociado con cada llave hay una lista de tuplas. El primer elemento de cada tupla es un string que contiene el nombre del libro, seguido de un caracter de punto y coma, seguido de la edición. El segundo elemento de cada tupla corresponde al precio de venta. Por ejemplo:

```
librerias = {"Ateneo": [("Ciencias;2ed", 10000), ("Papelucho;3ed", 3000), ("LyC;1ed", 5000),
                    ("Historia;1ed", 3000), ("English;2ed", 2000), ...],
            "Internacional": [("Ciencias;2ed", 8000), ("Papelucho;1ed", 5000),
                              ("Historia;2ed", 3000), ("English;3ed", 7500),
                              ("Matematica;1ed", 9000), ...],
            "Nacional": [("LyC;2ed", 4500), ("Ciencias;1ed", 9000), ("Papelucho;2ed", 6000),
                        ("Literatura;1ed", 3000), ("Historia;2ed", 3500),
                        ("Matematica;2ed", 9000), ...],
            ...}
```

Un mismo libro puede venderse en distintas librerías, al mismo o a distinto precio, y puede corresponder a distintas ediciones. Para los apoderados es indiferente la edición, pues lo único que les interesa es el menor precio. Por ejemplo, el libro 'LyC' es ofrecido por la Librería Ateneo en su primera edición, a un precio de 5000, pero también por la Librería Nacional en su segunda edición a un precio de 4500. **Suponga que una librería vende siempre una única edición de cada libro.**

- a) Implemente la función `buscar_libro(librerias, libro)`, que recibe como parámetro el diccionario con el inventario de las librerías y el nombre de un libro en particular, sin mención de edición. La función debe retornar un diccionario que contenga una entrada por cada librería que venda el libro en cuestión, en cualquier edición. La llave es el nombre de la librería y el valor es el precio al que se vende el libro en esa librería. **En caso de que el libro consultado no se encuentre en ninguna librería, retornar un diccionario vacío.** [15%]

```
>>> buscar_libro(librerias, 'LyC')
{'Ateneo':5000, 'Nacional':4500}

>>> buscar_libro(librerias, 'Biologia')
{}
```

- b) Implemente la función `donde_comprar(librerias, lista)`, que recibe como parámetro el diccionario con el inventario de las librerías, junto con una lista de libros que se necesita comprar. La función debe retornar un diccionario en el que existe una entrada por cada libro buscado. La llave es el nombre del libro y el valor corresponde a una tupla con el nombre de la librería donde conviene comprar y el precio que se deberá pagar. **En caso de que no se encuentre un libro en particular, la tupla correspondiente deberá estar compuesta por un string vacío y el valor 0.** Cuando haya varias alternativas con el menor precio, su solución puede utilizar cualquiera de ellas. [25%]

```
>>> libros = ["LyC", "Papelucho", "Matematica", "Biologia"]
>>> donde_comprar(librerias, libros)
{'Papelucho': ('Ateneo', 3000), 'Biologia': ('', 0),
 'Matematica': ('Internacional', 9000), 'LyC': ('Nacional', 4500)}
```

Programación – Cert. Recuperativo - Martes 18 de Diciembre de 2018

3. [40 %] Es fin de año y la revista Pyllboard debe publicar dos rankings en función de los videos que están en la plataforma PyTube: el grupo de las 10 canciones con más visitas y los 10 artistas con mayor cantidad de visitas sumando todas sus canciones. Para ello escribieron en un archivo todos los artistas a considerar. Cada línea contiene un artista diferente.

Por otro lado cada artista tiene su propio archivo de texto donde por cada línea está el título de una canción, un espacio en blanco y su respectiva cantidad de visitas, pero donde hay un punto cada 3 dígitos (de derecha a izquierda) para facilitar la lectura. Notar además que el nombre del archivo de texto de un artista es su nombre en minúsculas, sin considerar espacios en blanco, más la extensión “.txt”. A continuación se muestran ejemplos de los archivos (notar que no hay orden establecido para los datos):

artistas.txt

```
Pylor Pyft
Juan Pys Guerra
CalPyn Harris
Dua Pypa
...
```

calpynharris.txt

```
How Py is your love 89.000.331
This Is What You Py For 22.298.098
Pystroke 789.322
One Py 78.983.029
...
```

La revista lo contrata a usted para que elabore la función `rankings(arch_art)`, donde `arch_art` es un string que corresponde al nombre del archivo donde están todos los artistas (como “artistas.txt”). La función debe escribir los archivos de los dos rankings solicitados, donde por cada línea estará la posición (en orden ascendente), el nombre de la canción o artista y el número de visitas, separados por espacios en blanco. Suponiendo que “Pylor Pyft” es la artista con más visitas y la canción “How Py is your love” (que no necesariamente corresponde a “Pylor Pyft”) es la que tiene más visitas, al ejecutar el programa se deberían escribir los siguientes archivos y retornar la tupla (‘How Py is your love’, ‘Pylor Pyft’).

rank_canciones.txt

```
1 How Py is your love 89000331
2 Look what you Py do 88999002
...
10 PyLove 991020
```

rank_artistas.txt

```
1 Pylor Pyft 533994012
2 CalPyn Harris 191070780
...
10 Justin Pyber 8967
```

Notar que al escribir los números en estos archivos no se ponen puntos como separador.

El presidente de Pyllboard le indica además que no se preocupe por eventuales empates, simplemente debe ordenar según mayor cantidad de visitas y considerar hasta el décimo, sabiendo que siempre habrán muchos más. Recuerde que pueden haber miles de artistas y canciones.