



## Programación—C.Recuperativo (CC) - 30 de Agosto de 2017

Nombre

Rol

Paralelo

2. [40 %] Un Supermercado Independiente de Pythonia maneja la información sobre cargamentos y categorías de producto en las siguientes estructuras:

Cargamentos es un diccionario que maneja los lotes de artículos que llegarán en determinados días durante la siguiente semana (Lunes a Viernes).

```
cargamentos = {'Lunes' : {'bebidas', 'arroz', ...},
               'Martes' : {'agua_mineral', 'pescado', 'harina', ...},
               'Miercoles' : {'arroz', 'fideos', 'jugo', ...},
               'Jueves' : {'vasos', 'agua', 'queso', 'pan', ...},
               'Viernes' : {'bebida', 'queso', 'cerveza', 'pan', 'jugo', ...} }
```

El diccionario `categorias` contiene la categoría a las que pertenece cada artículo.

```
categorias = { 'bebestibles' : { 'bebida', 'cerveza', 'jugo', 'agua' ...},
               'abarrotes' : { 'queso', 'jamon', 'pan', 'sal', 'harina' ...},
               ...
               'alimentos' : { 'arroz', 'fideos', 'carne', 'pescado', ...} }
```

Considerando las variables `categorias` y `cargamentos`, como variables globales (que no deben ser modificadas), realice lo siguiente:

a) Escriba la función `recibos(dia)`, la cual recibe como parámetro un string con el nombre del día a consultar. La función debe retornar un diccionario cuyas llaves correspondan a las categorías que se reciben en el día especificado y cuyos valores corresponden a la **cantidad** de productos de cada categoría.

```
>>> recibos('Viernes')
{'abarrotes':2, 'bebestibles':3, ...}
```

b) Escriba la función `promocionar(dia, excluir)`, la cual recibe un string con el día de la semana a promocionar y un conjunto de categorías llamado `excluir`. La función debe retornar la categoría que más productos recibe dicho día, sujeto a que no se encuentre dentro del conjunto `excluir`. Asuma que el conjunto `excluir` tiene menor cardinalidad que las categorías de productos que se reciben dicho día y que en caso de empate, retorna cualquiera de ellas.

```
>>> promocionar('Lunes', set(['bebestibles']))
'alimentos'
```

c) Escriba la función `promo_semana()`, la cual no recibe parámetros. Esta función debe retornar una lista con las categorías a promocionar cada día. El primer elemento debe ser lo que se promocionará el día Lunes y así continuar hasta la promoción del día Viernes. **Los 5 días deben promocionar una categoría distinta**. Asuma que para cada día el cargamento tendrá productos de al menos 5 categorías distintas.

```
>>> promo_semana()
['bebestibles', 'alimentos', 'abarrotes', 'vestuario', 'aseo']
```

## Programación—C.Recuperativo (CC) - 30 de Agosto de 2017

Nombre

Rol — Paralelo

3. [40 %] Una empresa de productos ha comenzado a vender sus artículos fuera de Chile. Cada producto que será enviado, cuenta con un código único para poder ser identificado. Para que un producto llegue a destino, muchas veces debe realizar un viaje por distintos países antes de llegar al cliente. Esta información es guardada en el archivo *envios.txt*, donde cada línea tiene el formato *codigo:paisOrigen-pais2-...paisDestino*, donde siempre el primer país es el de origen y el último el de destino. Los países intermedios son aquellos por los cuales debe pasar para llegar a destino, notar que pueden existir más de un país intermedio.

envios.txt

```
RN1021:chile-peru-colombia-panama
NB812:chile-peru-ecuador-brasil
KM192:chile-peru-bolivia-brasil
UY291:chile-peru-australia
LA190:chile-argentina-uruguay-brasil
IW131:chile-panama
...
```

tiempos.txt

```
chile+peru=2$70000
colombia+peru=1$40000
chile+argentina=2$65000
peru+bolivia=2$30000
bolivia+brasil=3$50000
argentina+uruguay=1$25000
peru+ecuador=3$55900
uruguay+brasil=3$75000
peru+australia=5$103500
panama+chile=4$95000
panama+colombia=1$58400
ecuador+brasil=2$41000
...
```

Además, durante el tiempo se ha logrado conocer los días que demora cada tramo y el costo monetario. Para ello, se tiene el archivo *tiempos.txt*, donde cada línea tiene el formato *pais1+pais2=dias\$costo*. Considere que el costo de ir de *pais1* a *pais2* es el mismo en ambos sentidos. Asuma que en este archivo se encuentran todos los tramos posibles, pero se presentan en un sólo sentido, es decir, si está Chile y Perú no existe la línea *peru+chile=2\$70000*, en el archivo *tiempos.txt*

Desarrolle la función **generar\_reporte(maxd)** que recibe la máxima cantidad de días que se puede demorar un viaje desde el origen a destino considerando todos los tramos que debe realizar. La función debe crear un archivo llamado *X\_dias.txt*, donde *X* es la cantidad de días recibidos como parámetro. Este archivo debe guardar en cada línea el código del producto y los días que tardará en llegar en el formato *codigo:dias*.

Los viajes deben estar ordenados por el valor del envío. De más caro a más barato, aunque esta información no se debe escribir en el archivo. En caso de no existir viajes que cumplan con la condición, en el archivo se debe escribir el mensaje *No hay viajes para esta cantidad de días*

6\_dias.txt

```
RN1021:4
LA190:6
IW131:4
```

3\_dias.txt

```
No hay viajes con esta cantidad de días
```