PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2233 - PROGRAMACIÓN AVANZADA

Actividad 06

2^o semestre 2017
14 de Septiembre de 2017

Programación Funcional: Generadores y Built-ins

Introducción

La reina Barrios ha llegado al límite de popularidad que pudo conseguir en este universo teniendo a todos los seres vivientes en su lista de contactos. Sin embargo, aún no está satisfecha, por lo que el benevolente Nebil le propone viajar a la dimensión C-137 para buscar nuevas formas de vida que agregar a sus contactos.

Al llegar, se percató de que ese universo no era como el suyo. Por suerte, Lucas, el doglover, le envió archivos que le ayudarán a entender este nuevo entorno, en particular, Chile. Como Barrios es una reina ocupada, le pide a usted que trabaje con los archivos y entregue estadísticas utilizando programación funcional.

Objetivo

El objetivo de esta actividad es realizar un sistema de consultas utilizando **programación funcional** que permita obtener información sobre las personas, ciudades y países de este nuevo universo. Para realizar lo anterior, debe guardar la información de los archivos en una estructura de datos adecuada y completar las funciones entregadas en el archivo `AC06.py`.

Archivos

Contienen la información del nuevo universo. A continuación se detalla el contenido de los archivos:

- `Ciudades.txt`: Contiene la información de los ciudades en la dimensión. Cada fila es de la forma:

  `sigla país, nombre.`

- `Paises.txt`: Contiene la información de las países en la dimensión. Cada fila es de la forma:

  `sigla, nombre.`

- `Informacion_personas.txt`: Contiene la información de las personas en la dimensión. Cada fila es de la forma:

  `nombre, apellido, edad, sexo, ciudad de residencia, área de trabajo, sueldo.`

Funciones

Debe completar las funciones entregadas para responder las consultas posteriores.

- `ciudad_por_pais(nombre_pais, paises, ciudades)`
  Recibe el nombre de un país, una lista de paises y una lista de ciudades.
  Debe retornar un generador con las ciudades pertenecientes al pais X.
- `personas_por_pais(nombre_pais, paises, ciudades, personas)`
  Recibe el nombre de un país, una lista de paises, una lista de ciudades y una lista de personas.
  Debe retornar un generador con las personas pertenecientes al país X.
- `personas_por_ciudad(nombre_ciudad, personas)`
  Recibe el nombre de una ciudad y una lista de personas.
  Debe retornar un generador con las personas pertenecientes a la ciudad X.
- `personas_con_sueldo_mayor_a(personas, sueldo)`
  Recibe una lista de personas y un valor de sueldo.
  Debe retornar un generador con las personas con un sueldo mayor a X.
- `personas_por_ciudad_sexo(nombre_ciudad, sexo, personas)`
  Recibe el nombre de una ciudad, un sexo y una lista de personas.
  Debe retornar un generador con las personas de una ciudad X y de un sexo Y.
- `personas_por_pais_sexo_profesion(nombre_pais, paises, sexo, profesion,ciudades, personas)`
  Recibe el nombre de un país, una lista de paises, un sexo, una profesión, una lista de ciudades y una lista de personas.
  Debe retornar un generador con las personas de un pais X, sexo Y y profesión Z.
- `sueldo_promedio(personas)`
  Recibe una de lista personas.
  Debe retornar un valor (int/float).
- `sueldo_ciudad(nombre_ciudad, personas)`
  Recibe el nombre de una ciudad y una lista de personas.
  Debe retornar un valor (int/float).
- `sueldo_pais(nombre_pais, paises, ciudades, personas)`
  Recibe el nombre de un país, una lista de paises, una lista de ciudades y una lista de personas.
  Debe retornar un valor (int/float).
- `sueldo_pais_profesion(nombre_pais, paises, profesion, ciudades, personas)`
  Recibe el nombre de un país, una lista de paises, una profesión, una lista de ciudad y una lista de personas.
  Debe retornar un valor (int/float).

Ejemplos

Para saber si sus funciones se comportan de la manera esperada. Se darán ejemplos utilizando dichas funciones para casos particulares. **No modifique nada de los ejemplos.**

Consultas propias

Finalmente se te pide mostrar las siguientes consultas utilizando las funciones creadas en el punto anterior:

- Muestre a las 10 personas en Chile con mejor sueldo y **enumérelos** con índices partiendo desde el 0. (**hint:** use `enumerate`)
  Ejemplo para 3 personas:

  0, Nicolas, 4979
  1, Andrea, 4976
  2, Jan, 4963

- Se le pide, utilizando `zip` seleccionar 10 personas al azar en el mundo y generar tuplas con sus nombres, apellidos y sueldos.
  Ejemplo para 4 personas:

  Nombres: ('Lucas', 'Camilo', 'Camilo', 'Orlando')
  Apellidos: ('Aravena', 'Venezian', 'Garrido', 'Otarola')
  Sueldos: (121, 232, 612, 134)

Notas

- **Las namedtuple pueden resultarle útiles.**
- **No modifique los archivos .txt entregados.**
- **Queda prohibido el uso de for y while, excepto para generadores, listas y generadores por comprensión e imprimir resultados.**
- **Se recomienda FUERTEMENTE hacer la actividad en orden, puede que las funciones completadas le ayuden con una posterior.**
- **Asuma que los nombres de ciudades son únicos, es decir, el nombre de una ciudad no se repite en ningún otro país del mundo.**
- **NO suba los archivos .txt a su repositorio, el subir los archivos será severamente penalizado.**
- **Debe tener un archivo `.gitignore` en la carpeta `AC06`**

Requerimientos

- (0.8 pts) Hace buen uso de programación funcional

- (4.0 pts) Funciones
  - (0.4) `ciudad_por_pais`
  - (0.4) `personas_por_pais`
  - (0.4) `personas_por_ciudad`
  - (0.4) `personas_con_sueldo_mayor_a`
  - (0.4) `personas_por_ciudad_sexo`
  - (0.4) `personas_por_pais_sexo_profesion`
  - (0.4) `sueldo_promedio`
  - (0.4) `sueldo_ciudad`
  - (0.4) `sueldo_pais`
  - (0.4) `sueldo_pais_profesion`
- (1.2 pts) Consultas propias
  - (0.6) enumerados top 10 personas en Chile con mejor sueldo
  - (0.6) 10 personas al azar con mejor sueldo

Entrega

- **Lugar:** En su repositorio de Github en la **carpeta** `Actividades/AC06/`
- **Hora:** 16:55