

Programación – Certamen Rec. – Semestre 2, 2017

PARALELO

NOMBRE

ROL -

2.

Programación – Certamen Rec. – Semestre 2, 2017

PARALELO

NOMBRE

ROL -

3.

Programación – Certamen Rec. – Semestre 2, 2017

2. [40%] TurismPy desea conocer información turística del país. Para ello dispone de un diccionario cuya llave corresponde al lugar turístico y el valor es una lista de tuplas, donde cada tupla contiene: cantidad de turistas (en miles), código del país de origen y divisas dejadas en ese lugar turístico en moneda origen (en miles):

```
turismo = { 'San Pedro':[(1200, 'Arg', 150), (53, 'Fr', 240)],
            'Valparaiso':[(1500, 'EEUU', 23)],
            'Isla Pascua':[(45, 'Fr', 46), (77, 'SZ', 63), (200, 'Arg', 680)],
            'Chiloe':[(119, 'Bra', 80), (50, 'Pe', 350)], ...}
```

Además se tiene un conjunto de tuplas, donde cada tupla contiene el código del país de origen, el nombre del país y su moneda:

```
países = (('Arg', 'Argentina', 'peso A'), ('EEUU', 'Estados Unidos', 'dolar'),
          ('Fr', 'Francia', 'euro'), ('SZ', 'Suiza', 'franco'),
          ('Bra', 'Brasil', 'real'), ('Pe', 'Peru', 'sol'), ...)
```

Finalmente se tiene un diccionario con el factor a multiplicar de cada moneda para obtener el equivalente en moneda nacional, es decir, peso chileno.

```
monedas = {'dolar' : 620, 'peso A' : 36, 'sol' : 191,
           'franco' : 641, 'real' : 198, 'euro' : 739, ...}
```

Considerando que pueden existir muchos otros lugares turísticos y países:

- a) Escriba la función `lugar_mas_visitado(turismo)`, que recibe como parámetro el diccionario `turismo` y retorna el lugar que ha sido más visitado por turistas.

```
>>> lugar_mas_visitado(turismo)
'Valparaiso'
```

- b) Escriba la función `divisas_pais(turismo)`, que recibe como parámetro el diccionario `turismo` y retorna un diccionario cuya llave es el código del país y el valor es el total de divisas de dicho país. Recuerde multiplicar por 1000 los valores. Guíese por el ejemplo.

```
>>> divisas_pais(turismo)
{'Arg': 830000, 'EEUU': 23000, 'Fr': 286000,
 'SZ': 63000, 'Bra': 80000, 'Pe': 350000}
```

- c) Escriba la función `total_divisas(turismo,monedas,países)`, que recibe como parámetro los diccionarios `turismo` y `monedas`, y el conjunto `países`, retornando el total en divisas en moneda nacional.

```
>>> total_divisas(turismo,monedas,países)
378567000
```

Programación – Certamen Rec. – Semestre 2, 2017

3. [40 %] En el año 28908 hay elecciones municipales en Pythonlandia y cada mesa de votación cuenta con una urna sumamente tecnológica. Esta urna escanea el voto depositado y escribe la información en un archivo siguiendo el formato `rut:alcalde-concejaj`, donde `rut` es el rut de la persona que debe estar inscrita en esta mesa, `alcalde` es el código del alcalde y `concejaj` el código de concejal por los que vota dicha persona. Ver ejemplo: `votos.txt`. Además las personas que están inscritas en esta mesa se encuentran en el archivo `electores.txt` donde por cada línea existe un rut (y no hay repetidos).

Como ya no existen vocales de mesa, una persona podría votar más de una vez y esto quedará registrado en el archivo. Por este motivo solo se debe considerar la primera votación efectuada. En otras palabras, si el rut se repite, simplemente no se consideran las nuevas votaciones (la primera sí se considera válida). Además, no existen los votos en blanco o nulos. Toda persona vota exactamente por un alcalde y un concejal.

Considere que la información de los candidatos a Alcalde y Concejal se encuentran en los diccionarios `alcaldes` y `concejales` como se muestra a continuación:

```
alcaldes = {'A1': 'Alejandro Pinera',
            'A2': 'Sebastian Guillier',
            'A3': 'Beatriz Kast', ...}
```

```
concejales = {'C1': 'Leopoldo Sharp',
              'C2': 'Jorge Mendez', ...}
```

Asuma que los votos (del archivo `votos.txt`) solo tendrán candidatos que existen en estos dos diccionarios.

Cree la función `resultados(vot, ele, alc, con)`, donde `vot` y `ele` son strings con los nombres de los archivos que continene los votos y los electores de esta mesa respectivamente y `alc` y `con` corresponden a los diccionarios de alcaldes y concejal respectivamente, ya descritos. Esta función debe generar el archivo `resultados.txt`, cuyo encabezado es el total de votos válidamente emitidos, es decir, eliminando los votos de personas que no sean de esta mesa y sin contar repetidos. Luego están los resultados para Alcalde y finalmente para Concejal. Guíese por el ejemplo (`resultado.txt`).

El orden en que aparecen los candidatos en el archivo `resultado.txt` es irrelevante y si un candidato (alcalde o concejal) no obtuvo votos, simplemente se omite.

Nota: considere que todos los archivos tienen más datos. Además como el voto es secreto, usted debe eliminar el archivo que permite identificar cómo votó cada persona. La función `os.remove(nom_arch)` recibe un string `nom_arch` y borra el archivo correspondiente a dicho nombre.

`votos.txt`

```
9173315-3:A1-C2
19642628-0:A3-C1
21933617-9:A2-C1
9173315-8:A2-C1
...
```

`electores.txt`

```
9173315-3
19642628-0
...
```

`resultado.txt`

```
Total de votos: 2

Votacion alcaldes

Alejandro Pinera 50.0%
Beatriz Kast 50.0%
...

Votacion concejales

Jorge Mendez 50.0%
Leopoldo Sharp 50.0%
...
```