



## Programación—Certamen 2 (CSJ y CV) - Jueves 5 de Junio de 2014

Nombre:  Rol: -

2. [35 %] Una prestigiosa línea aérea nacional programa sus vuelos mediante el uso de la lista de tuplas `vuelos`, donde cada tupla corresponde a un vuelo distinto. Dentro de cada tupla, se ingresa el número del vuelo y su fecha de salida, tal como se muestra en el siguiente **ejemplo**:

```
vuelos = [(10, (2014,01,02)), (11, (2014,01,02)), (12, (2014,01,03)),  
(13, (2014,05,01)), (14, (2014,05,01))]
```

Por otra parte, cada vuelo puede tener varios destinos, lo cual se almacena en el diccionario `destinos` donde la llave es el número del vuelo y los datos asociados corresponden a un conjunto de tuplas de destinos. Cada destino tiene la forma `(Ciudad, País)`, tal como se muestra en el siguiente ejemplo:

```
destinos = {  
10: {('Lima', 'Peru'), ('San Jose', 'Costa Rica'), ('Los Angeles', 'USA')},  
11: {('San Jose', 'Costa Rica'), ('C. de Panama', 'Panama')},  
12: {('Sao Paulo', 'Brasil'), ('San Jose', 'Costa Rica')},  
13: {('Lima', 'Peru'), ('San Jose', 'Costa Rica'), ('C. de Panama',  
      'Panama')},  
14: {('San Jose', 'Costa Rica'), ('Buenos Aires', 'Argentina')}  
}
```

Considere las variables `vuelos` y `destinos` como globales.

- a) Desarrolle la función `vuelos_a_destino(destino, fecha)` que retorne una lista de vuelos que salen a un destino en una fecha dada.

```
>> vuelos_a_destino(('San Jose', 'Costa Rica'), (2014,05,01) )  
[13, 14]
```

- b) Desarrolle la función `destinos_repetidos()` que retorne un conjunto de tuplas con los destinos que figuran en todos los vuelos.

```
>>> destinos_repetidos()  
set([('San Jose', 'Costa Rica')])
```

- c) Desarrolle la función `paises_visitados(fecha)` que retorne el conjunto de países que tienen vuelo programado para una fecha dada.

```
>>> paises_visitados((2014,05,01))  
set(['Argentina', 'Panama', 'Costa Rica', 'Peru'])
```

## Programación—Certamen 2 (CSJ y CV) - Jueves 5 de Junio de 2014

Nombre:  Rol: -

3. [35 %] Se lleva a cabo un campeonato de atletismo interuniversitario el que consiste en un conjunto de pruebas en pista y otras en campo en categorías de damas (d) y varones (v). Esta información se encuentra almacenada en un diccionario cuyos valores son conjuntos, llamado `competencia`, como ejemplo 100 metros varones se registra como `'v100mt'`.

```
competencia={
  'pista': {'v100mt', 'v400mt', 'v800mt', 'v3000mt', 'd100mt', 'd400mt', ...},
  'campo': {'vbala', 'vdisco', 'vlargo', 'dbala', ...} }
```

El puntaje que se le otorga al deportista corresponde al rendimiento en la prueba y según ese puntaje tiene un lugar en la competencia. El diccionario `puntaje` tiene como llave el lugar en la competencia y su valor es la cantidad de puntos.

```
puntaje={'lugar 1': 12, 'lugar 2': 9, 'lugar 3': 7, 'lugar 4': 5,
        'lugar 5': 4, 'lugar 6': 3, 'lugar 7': 2, 'lugar 8': 1}
```

El primer lugar recibe medalla de oro, el segundo plata y el tercero bronce. Si nadie obtiene los primeros puntajes no hay medallas.

Por último se tiene el diccionario `resultado` que contiene la universidad (llave) y una lista con tuplas con el nombre de los deportistas, las pruebas y los puntajes que obtuvieron.

```
resultado={
  'usm': [ ('mrios', 'v400mt', 9), ('nmassu', 'v3000mt', 12),
          ('jrojas', 'vdisco', 12) ],
  'usach': [ ('jramos', 'd400mt', 5), ('lsoto', 'd400mt', 9),
            ('mruiz', 'v800mt', 7) ],
  'uc': [ ('mhard', 'v100mt', 3), ('msolis', 'd3000mt', 5),
          ('lrozas', 'dbala', 5) ], }
```

- a) Desarrolle la función `participante_prueba(competencia, resultado, prueba)` que recibe el diccionario de competencia, el diccionario de resultado y la prueba. La función debe retornar una lista con los nombres de los deportistas que compitieron en las pruebas indicadas.

```
>>> participante_prueba(competencia, resultado, 'campo')
['jrojas', 'lrozas']
```

- b) Desarrolle la función `mayor_cantidad(resultado, puntaje)` que reciba como parámetro el diccionario resultado y el de puntajes, y retorne la universidad que obtuvo el mayor número de medallas. Si dos universidades empatan, indique cualquiera.

```
>>> mayor_cantidad(resultado, puntaje)
'usm'
```

- c) Desarrolle la función `prueba_sin_medallas(resultado, puntaje)` que retorne una lista con las pruebas donde no hubo medallas.

```
>>> prueba_sin_medallas(resultado, puntaje)
['v100mt', 'd3000mt', 'dbala']
```