



## Algoritmos y Programación I (75.40 - 95.14) – Wachenchauzer – 2.º parcialito – 13/10/2017

Resolvé los siguientes problemas en forma clara y legible, respetando sangrías e incluyendo la documentación necesaria. Si te parece que los comentarios no son suficientemente explicativos, podés agregar una descripción de funcionamiento del código. Podés escribir tantas funciones auxiliares como creas necesarias.

1. Escribir una función que reciba una cadena y devuelva una tupla con la palabra que tuvo mayor cantidad de apariciones, y la cantidad de apariciones que tuvo. Si hay dos o más palabras con máxima cantidad de apariciones, devolver cualquiera de ellas. La cadena contiene únicamente palabras y espacios. Ejemplo: "la cama la silla y la mesa." -> ("la", 3).
2. Se cuenta con un archivo en formato `csv` que guarda información de pasajes de avión, respetando la siguiente estructura: `fecha,destino,precio`. Escribir una función que dada la ruta del archivo, devuelva un diccionario cuyas claves sean cada uno de los destinos, y el valor asociado a cada clave una tupla (`fecha`, `precio`) con el pasaje más barato para el destino.
3. Se quiere modelar un perchero para colgar ropa. Se pide crear las clases `Perchero` y `Prenda` tal que se se puedan ejecutar las siguientes líneas de código y se obtengan los resultados especificados. El constructor de `Perchero` recibe la cantidad de espacio total disponible, y el de `Prenda` recibe el nombre de la prenda y cuánto espacio ocupa:

```
>>> p = Perchero(3)
>>> p.colgar(Prenda('camisa', 1))
>>> p.colgar(Prenda('pantalon', 1))
>>> p.sacar('pantalon')
Prenda('pantalon', 1)
>>> p.sacar('remera')
Exception: No se encontró la prenda
>>> p.espacio_disponible()
2
>>> p.colgar(Prenda('campera', 3))
Exception: No hay espacio para colgar la prenda
¡Suerte! :-)
```



## Algoritmos y Programación I (75.40 - 95.14) – Wachenchauzer – 2.º parcialito – 13/10/2017

Resolvé los siguientes problemas en forma clara y legible, respetando sangrías e incluyendo la documentación necesaria. Si te parece que los comentarios no son suficientemente explicativos, podés agregar una descripción de funcionamiento del código. Podés escribir tantas funciones auxiliares como creas necesarias.

1. Escribir una función que reciba una cadena y devuelva una tupla con la palabra que tuvo mayor cantidad de apariciones, y la cantidad de apariciones que tuvo. Si hay dos o más palabras con máxima cantidad de apariciones, devolver cualquiera de ellas. La cadena contiene únicamente palabras y espacios. Ejemplo: "la cama la silla y la mesa." -> ("la", 3).
2. Se cuenta con un archivo en formato `csv` que guarda información de pasajes de avión, respetando la siguiente estructura: `fecha,destino,precio`. Escribir una función que dada la ruta del archivo, devuelva un diccionario cuyas claves sean cada uno de los destinos, y el valor asociado a cada clave una tupla (`fecha`, `precio`) con el pasaje más barato para el destino.
3. Se quiere modelar un perchero para colgar ropa. Se pide crear las clases `Perchero` y `Prenda` tal que se se puedan ejecutar las siguientes líneas de código y se obtengan los resultados especificados. El constructor de `Perchero` recibe la cantidad de espacio total disponible, y el de `Prenda` recibe el nombre de la prenda y cuánto espacio ocupa:

```
>>> p = Perchero(3)
>>> p.colgar(Prenda('camisa', 1))
>>> p.colgar(Prenda('pantalon', 1))
>>> p.sacar('pantalon')
Prenda('pantalon', 1)
>>> p.sacar('remera')
Exception: No se encontró la prenda
>>> p.espacio_disponible()
2
>>> p.colgar(Prenda('campera', 3))
Exception: No hay espacio para colgar la prenda
¡Suerte! :-)
```