

P3.3 - Ejercicios: Conjuntos

Ejercicio 3.3.1

Suponer una lista con datos de las compras hechas por clientes de una empresa a lo largo de un mes, la cual contiene tuplas con información de cada venta: (cliente, día del mes, monto, domicilio del cliente). Ejemplo:

```
[("Nuria Costa", 5, 12780.78, "Calle Las Flores 355"), ("Jorge Russo", 7, 699, "Mirasol 218"), ("Nuria Costa", 7, 532.90, "Calle Las Flores 355"), ("Julián Rodríguez", 12, 5715.99, "La Mancha 761"), ("Jorge Russo", 15, 958, "Mirasol 218")]
```

Escribir una función que reciba como parámetro una lista con el formato mencionado anteriormente y retorne los domicilios de cada cliente al cual se le debe enviar una factura de compra. Notar que cada cliente puede haber hecho más de una compra en el mes, por lo que la función debe retornar una estructura que contenga cada domicilio una sola vez.

Ejercicio 3.3.2

Solicitar al usuario que introduzca los nombres de pila de los alumnos de primaria de una escuela, finalizando cuando se introduzca “x”. A continuación, solicitar que introduzca los nombres de los alumnos de secundaria, finalizando al introducir “x”.

- Mostrar los nombres de todos los alumnos de primaria y los de secundaria, sin repeticiones.
- Mostrar qué nombres se repiten entre los alumnos de primaria y secundaria.
- Mostrar qué nombres de primaria no se repiten en los de nivel secundaria.
- Mostrar si todos los nombres de primaria están incluidos en secundaria.

Ejercicio 3.3.3

El conjunto potencia de un conjunto S es el conjunto de todos los subconjuntos de S .

Por ejemplo, el conjunto potencia de $\{1, 2, 3\}$ es:

```
{ $\emptyset$ , {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}}
```

Escriba la función `conjunto_potencia(s)` que reciba como parámetro un conjunto cualquiera s y retorne su «lista potencia» (la lista de todos sus subconjuntos):

```
>>> conjunto_potencia({6, 1, 4})
[set(), set([6]), set([1]), set([4]), set([6, 1]), set([6, 4]), set([1, 4]), set([6, 1, 4])]
```

Ejercicio 3.3.4

Dadas las siguientes listas:

```
frutas1 = ["manzana", "pera", "naranja", "plátano", "uva"]
frutas2 = ["manzana", "pera", "durazno", "sandía", "uva"]
```

1. Crea conjuntos a partir de estas listas y nómbralos `set_frutas1` y `set_frutas2`.
2. Encuentra las frutas que están en ambas listas y guárdalas en un nuevo conjunto llamado `frutas_comunes`.
3. Encuentra las frutas que están en `frutas1` pero no en `frutas2` y guárdalas en un conjunto llamado `frutas_solo_en_frutas1`.
4. Encuentra las frutas que están en `frutas2` pero no en `frutas1` y guárdalas en un conjunto llamado `frutas_solo_en_frutas2`.
5. Encuentra las frutas que están en `frutas 1` pero no en `frutas 2` y los que están en `frutas 2` pero no están en `frutas 1`. Guárdalas en un conjunto llamado `frutas_raras`

Ejercicio 3.3.5

Dado el conjunto de números enteros:

```
numeros = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

1. Crea un conjunto `pares` que contenga los números pares del conjunto `numeros`.
2. Crea un conjunto `multiplos_de_tres` que contenga los números que son múltiplos de tres del conjunto `numeros`.
3. Encuentra la intersección entre los conjuntos `pares` y `multiplos_de_tres` y guárdala en un conjunto llamado `pares_y_multiplos_de_tres`.
4. Indica si el conjunto `números` es un superconjunto de `pares`.

Ejercicio 3.3.6

Dado el conjunto de letras:

```
vocales = {'a', 'e', 'i', 'o', 'u'}
```

1. Crea un conjunto `consonantes` que contenga las letras del alfabeto que no son vocales.
2. Crea un conjunto `letras_comunes` que contenga las letras que están tanto en el conjunto `vocales` como en el conjunto `consonantes`.
3. Crea un subprograma que dado cualquier conjunto de este ejercicio sea capaz de escribirlo en la pantalla. (Si lo has creado antes, reutilízalo)