

# CC1002 Introducción a la Programación

## Auxiliar 13

Prof. Benjamin Bustos  
Auxs. Cristobal Sepúlveda, Franco Sepúlveda

Fecha: 22 de noviembre de 2017

Resolver los problemas usando Python, utilizando estrictamente la Receta de Diseño entregada a lo largo del curso. Use nombres apropiados para funciones y variables, y realice testing cada vez que sea posible.

### Relaciones entre conjuntos

Una relación  $R$  entre dos conjuntos  $A$  y  $B$  es un conjunto de pares  $(x, y)$  con  $x$  en  $A$  e  $y$  en  $B$ . Por ejemplo, si  $A = \{b, c, a\}$  y  $B = \{2, 1\}$  entonces una relación  $R$  podría ser  $\{(b, 2), (c, 1), (b, 1)\}$ . Al respecto, y siguiendo la receta de diseño (incluyendo el testing), escriba las siguientes funciones considerando que un par de valores de una relación se representa con la estructura definida por `estructura.crear('par', 'x y')`.

- A) Escriba la función `imagen` que reciba una relación  $R$  y un valor  $X$  y entregue como resultado una lista recursiva con todos los valores  $Y$  que están relacionados con el valor  $X$ , es decir, una lista con todos los  $Y$  para los que existe un par  $(X, Y)$ . Nota. Una relación  $R$  se representa por una lista recursiva. Por ejemplo, si `R=lista(par('b',2), lista(par('c',1), lista(par('b',1),None)))` entonces la invocación `imagen(R,'b')` debe entregar `lista(2,lista(1,None))`.
- B) Escriba la función `preImagen` que reciba una relación  $R$  y un valor  $Y$  y entregue como resultado una lista de Python con todos los valores  $X$  que se relacionan con  $Y$ , es decir, una lista con todos los  $X$  para los que existe un par  $(X, Y)$ . Nota. Una relación  $R$  se representa por una lista de Python. Por ejemplo, si `R=[par('b',2),par('c',1),par('b',1)]` entonces la invocación `preImagen(R,1)` debe entregar `['c', 'b']`.

## Tabla periódica

La tabla periódica de los elementos químicos se puede manejar a través de las instrucciones indicadas en el siguiente ejemplo:

```
T=Tabla() # crea tabla T inicialmente sin ningún elemento químico
T.agregar('H', 1.00794) # agrega a T el elemento de símbolo H y
                        # de peso 1.00794 (si el símbolo ya
                        # existe, el peso se reemplaza)
T.peso('H') # entrega peso del elemento de símbolo H de la tabla T
            # (si el símbolo no existe entrega cero)
```

- A) Escriba la función `pesoMolecula` que reciba la tabla periódica (un objeto de la clase `Tabla` que ya contiene el símbolo y el peso de todos los elementos químicos) y una lista (de Python) que representa una molécula (por ejemplo  $H_2O$  se representa por la lista `['H',2], ['O',1]]`). La función debe calcular y entregar el peso de la molécula. Por ejemplo, el peso de  $H_2O$  se calcula como  $2*1.00794 + 1*15.9994$  (en que 1.00794 es el peso del Hidrógeno y 15.9994 el del Oxígeno). Nota. Si la molécula contiene un elemento que no existe en la tabla, la función debe entregar como resultado el valor cero.
- B) Siguiendo la receta de diseño para clases (incluyendo el testing), implemente la clase `Tabla` con todos sus métodos. Suponga que un objeto de la clase `Tabla` se representa por un diccionario en que la llave es el símbolo del elemento y el valor es su peso atómico. Por ejemplo:

```
self.__D = { 'H' : 1.00794,  'O' : 15.9994, ... }
```