

Aplicaciones con Lenguajes de Script – Parcial 1

Apellidos y nombre:

DNI:

e.mail:

1. (2 pts.) Implemente la función *elementos_comunes(l1, l2)*. Se trata de una función que recibe dos listas, y devuelve una nueva lista con aquellos elementos de *l1* que también están en *l2*. Abstenerse de utilizar conjuntos, o alguno de sus operadores.
2. (2 pts.) Implemente la función *histograma()*. Se trata de una función que recibe una lista de números, y visualiza en una línea por elemento, con una sucesión de asteriscos según el valor correspondiente. Será necesario normalizar los valores de la lista, tomando el valor posible más bajo 0, y de forma que como mínimo no se imprimirá ningún asterisco, y como máximo, diez. Nótese que para los valores [100, 50, 80], se imprimirán, respectivamente, 10, 5 y 8 asteriscos, exactamente igual que para los valores [10, 5, 8], o [20, 10, 16].
3. (2 pts.) Una función *es_subconjunto(c1, l1)* admite un conjunto *c1* y una lista de conjuntos *l1*. En caso de que *l1* o *c1* estén vacíos o sean *None*, la función devuelve *False*. En cambio, si *c1* es un subconjunto de todos los conjuntos de *l1*, devuelve *True*. Resuelva el problema utilizando obligatoriamente conjuntos y funciones anidadas.
4. (2 pts.) Cree la clase **TableroTetris**. La clase guardará una lista de listas para representar las posiciones del tablero, que son fijas, concretamente 10 de ancho por 17 de alto. Tendrá un constructor que inicializará el tablero a vacío (cada casilla con un espacio, en contraposición a la casilla ocupada, que tendrá una 'X'); el método `__str__` que volcará el tablero por pantalla; el método *set(f, c, b)*, que modificará la posición dada (*f*: fila, *c*: columna) con una 'X' o un espacio según el valor booleano *b*; *get(f, c)* que devolverá un booleano según la casilla esté ocupada o no; y finalmente *elimina_filas()*, que eliminará aquellas filas *n* que estén llenas (toda la fila tiene 'X' como valor de columna), moviendo todas las filas superiores a *n* hacia abajo, y creando una nueva fila vacía en la parte superior del tablero.
5. (2 pts.) Cree la clase **Figura**. Cree las clases **Círculo** y **Rectángulo** derivando de **Figura**. Todas las figuras tienen una posición determinada por un **Punto**. Cree la clase **Dibujo**, que guarda una colección de figuras a través de un método *add(l)*. La clase **Círculo** guarda un radio y un **Punto** con su posición. La clase **Rectángulo** guarda un ancho, un alto y el **Punto** que determina su vértice superior izquierdo. Cree métodos `__str__()` en todas las clases, así como propiedades adecuadas para cada atributo. Nótese que los constructores de **Círculo** y **Rectángulo** admiten tuplas u objetos **Punto** para la posición inicial. También calculan el **Punto** más alejado desde la posición inicial.

```
> d = Dibujo()
> d.add([Circulo(Punto(10, 10), 1), Rectangulo((20, 20), 10, 5)])
> print(d)
[Circulo(10): (10, 10)-(20, 20) | Rectangulo(10, 5): (20, 20)-(30, 25)]
```