

Profesor: Dr. Oldemar Rodríguez Rojas

Paradigmas de Programación

Programación en Lenguaje Python

Fecha de Entrega: **jueves 9 de junio a las 8am**

TAREA NÚMERO 9

- Las tareas son estrictamente de carácter individual, tareas idénticas se les asignará cero puntos.
 - Todas las tareas tienen el mismo valor en la nota final del curso, es decir, el promedio de las notas obtenidas en la tareas será la nota final del curso.
 - Todas las preguntas tienen el mismo puntaje.
 - Incluir al menos 3 casos de prueba en las funciones programadas (cuando corresponda).
1. Programe todos los ejercicios de la tareas 1 y 2, pero esta vez usando el lenguaje **Python** con un estilo 100 % funcional (es decir, replique en lenguaje **Python** las soluciones realizadas en **Scheme**).

Para esto debe seguir las siguientes reglas:

- a) NO puede usar ningún tipo de bucle (ciclo), es decir, no puede usar `for` En este tipo de ejercicios se debe usar recursión o recursión lineal según se indica en la Tarea 1 o Tarea 2.
- b) NO debe usar asignación, excepto en el ejercicio 5 y 6 de la Tarea 1, en donde se requería del uso del `let`.
- c) En los ejercicios que requieren el uso de `cons`, `car` o `cdr` debe usar el paquete `cons`. Para instalarlo use `pip install cons` en la consola de **Anaconda** o en la consola de **macOS**. Por ejemplo:

```
from cons import cons, car, cdr
```

```
L = [1, 4, 5, -8, 99]
```

```
L
```

```
Out[14]: [1, 4, 5, -8, 99]
```

```
car(L)
```

```
Out[15]: 1
```

```
cdr(L)
```

```
Out[16]: [4, 5, -8, 99]
```

```
cdr(cdr(L))
```

```
Out[18]: [5, -8, 99]
```

d) Para funciones anónimas se tiene la siguiente equivalencia (por ejemplo):

En **Scheme**: `lambda(x) (x+2)`

En **Python**: `sumar = lambda x: x+2`

Entregables: Debe entregar un documento autreproducible HTML con todos los códigos y salidas.



oldemar **rodríguez**

CONSULTOR en MINERÍA DE DATOS