

TRABAJO PRÁCTICO #4

Introducción a listas en Python

Condiciones de Entrega

Los trabajos deberán ser enviados al correo proa.prog1@gmail.com especificando como asunto el número de grupo. Todos los ejercicios enviados después de las 15:00 hs del 29/09 tendrán la mitad de puntaje

Ejercicio 1 (1 pto)

Leer 10 valores enteros y almacenarlos en una lista. Mostrar el elemento que ocupa la primer posición y luego mostrar el promedio de todos los elementos de la lista.

Ejercicio 2 (1 pto)

Leer valores hasta que se ingrese 0 y almacenarlos dentro de una lista. Luego mostrar la lista y su tamaño.

Ejercicio 3 (1.5 ptos)

Leer 10 valores enteros y almacenarlos en una lista. Mostrar la lista completa. Luego pedir al usuario que ingrese una opción: 1) Mostrar sólo los elementos en las posiciones pares o 2) Mostrar sólo los elementos de las posiciones impares. Mostrar los elementos de la lista correspondientes a la opción que el usuario eligió.

Ejercicio 4 (1.5 pts)

Se leen las notas de una evaluación de *Programación I* correspondientes a 10 alumnos. El programa debe informar la cantidad de notas que están por encima del promedio.

Ejercicio 5 (1.5 ptos)

Leer 10 valores enteros y almacenarlos en una lista, luego pedir al usuario que ingrese otro valor numérico y contar y mostrar cuántas veces aparece dicho valor dentro de la lista.

Ejercicio 6 (1.5 ptos)

Leer 10 valores enteros y almacenarlos en una lista, luego buscar y mostrar **cuál es el mayor valor** y **qué posición ocupa** dentro de la lista.

Ejercicio 7 (2 pts)

Un sensor registró las temperaturas durante el mes de septiembre tomando una medición por día. El sensor tenía un defecto de fabricación, por lo cual algunas mediciones de temperatura resultaron defectuosas y fueron almacenadas con el valor (-1). Leer los valores tomados por el sensor y almacenarlos en una lista, luego corregir las mediciones reemplazando los valores (-1) por el promedio de sus vecinos. Por ejemplo, si la lista original fue:

```
[ 1 2 -1 3 4 -1 5]
```

La lista corregida debería quedar:

```
[ 1 2 2.5 3 4 4.5 5]
```

Se sabe que no hay dos valores inválidos seguidos y que el primero y último valor de la lista son válidos. Mostrar la lista original y la lista corregida.

Ayudín...

- Crear una lista vacía:

```
a = []
```

- Agregar un elemento (ej: el entero 3):

```
a.append(3)
```

- Obtener el tamaño de la lista

```
len(a)
```

- Mostrar el elemento en una determinada posición (ej: el segundo elemento):

```
i = 1  
print a[i]
```

Atención: recordar que los índices de los elementos empiezan desde 0, por lo que la posición 1 contiene al segundo elemento

- Modificar el elemento en una posición dada (ej: cambiar el elemento en la primer posición por un 3)

```
pos = 0  
a[pos] = 3
```