

Instrucciones:

- Para todos los problemas identifique los datos de entrada, los datos de salida y las operaciones necesarias sobre los datos de entrada para obtener los datos de salida.
- Para todos los programas utilice la siguiente estructura:

```
#CONSTANTES
instrucciones...
#ENTRADAS
instrucciones...
#PROCESAMIENTO
instrucciones...
#SALIDAS
instrucciones...
```

- Comente cada una de las líneas de los programas.
- Siga las buenas prácticas. Está prohibido utilizar Inteligencia Artificial para resolver u obtener una “ayuda” con los problemas (ChatGPT, Gemini, Bing, Github Copilot, entre otros).
- Para cada problema haga un archivo .py.
- El orden de los ejercicios no representa la dificultad. Usted puede resolver los ejercicios en el orden que prefiera.

Ayuda:

- Operadores aritméticos: +, -, *, /, //, %, **
- Operadores de comparación: >, >=, <, <=, ==, !=
- Operadores lógicos: and, or, not
- Funciones nativas: int(), float(), str(), bool(), list(), print(), input(), range(), round(), len(), eval()
- Estructuras de control: if, if-else, if-elif-else, while, for-in
- Métodos de Strings: upper(), lower(), count(), isdigit(), isalpha(), isupper(), islower()
- Operador para saber si un String está contenido en otro: in
- Métodos de listas: append(), pop(), remove(), index(), insert(), copy()

Problema 1

Escriba un programa que permita crear una lista de palabras. Primero, el programa debe solicitar el número de palabras que tendrá la lista y luego solicitar el mismo número de palabras para crear la lista. Por último, el programa debe mostrar por consola cada palabra de la lista.

Problema 2

Implemente un programa que reciba una lista números enteros y un número entero como entrada e indique si el número ingresado se encuentra o no en la lista.

Recomendación: puede utilizar la función nativa `eval()` para ingresar la lista de números.

Problema 3

Escriba un programa que pida al usuario que introduzca una lista (no vacía) de números no necesariamente ordenada y que busque el mayor y el menor número que se ingresó. Además, debe mostrar el promedio de los elementos de la lista.

Ejemplo:

```
listaNumeros = [12,4,34,1,2,52,1,54,98,5,235,76,57,87,123,32,32]
```

```
menor = 1
```

```
mayor = 235
```

```
promedio = 53.2352941176
```

Problema 4

Construya un programa que tenga como entrada una lista de números e informe si todos los elementos de la lista son distintos o no.

Problema 5

Escriba una programa en lenguaje Python que indique si en una lista de n números solicitada al usuario se encuentran todos los números desde el número uno hasta el número n (no importa el orden).

Restricción: no se puede ordenar la lista.

Problema 6

Escriba un programa en Python que pida al usuario que introduzca una lista de números enteros positivos, y que luego imprima el número impar más grande que se ingresó. En caso de que se ingresen solo números pares, debe mostrar un mensaje indicando que no existen números impares.

Problema 7

Construya un programa que encuentre el número que más se repita en una lista de números solicitada al usuario. En caso de ser más de un número el que más se repita, el programa debe entregar una lista con todos ellos.

Construya el programa sin usar los métodos `.count()`, `.sort()` y `.index()`

Problema 8

Construya un programa en Python que solicite al usuario una lista de números enteros y entregue como salida una lista de números sin repeticiones.

Ejemplo:

Entrada: [4,3,1,2,4,3,5,9,5,2,3]

Salida: [4,3,1,2,5,9]

Problema 9

Implemente un programa en Python que reciba como entrada una lista de largo arbitrario con el siguiente formato:

`[[int, int, int, int], [[int, int, int, int]], ... , [int , int, int, int]]`

y entregue como salida la posición de la sublista con el mayor promedio.

Problema 10

Escriba un programa que reciba como entrada dos listas de números enteros y que entregue una lista con los números comunes de las listas (intersección). En caso de no existir elementos comunes, el programa debe informarlo.

Problema 11

Escriba un programa que reciba como entrada dos listas de números enteros y que entregue una lista con la unión de ambas listas, es decir una lista con todos los valores de ambas listas ingresadas, pero sin repeticiones.