

## Ejercicios prácticos (Listas)

- 1) Definir una lista que almacene 5 enteros. Sumar todos sus elementos y mostrar dicha suma.
- 2) Definir una lista que almacene los nombres de los primeros seis meses del año. Mostrar el primer y último elemento de la lista solamente.
- 3) Definir una lista que almacene como primer elemento el nombre de un alumno y en las dos siguientes sus notas. Imprimir luego el nombre y el promedio de las dos notas.
- 4) Definir una lista con 7 elementos enteros. Contar cuántos de dichos valores almacenan un valor superior a 33.
- 5) Definir una lista con 7 elementos enteros. Mostrar por pantalla solo los elementos con valor iguales o superiores a 5.
- 6) Definir y cargar una lista con 10 números enteros aleatorios (utilizar random), entre 1 y 10. Contar y mostrar por pantalla la cantidad de números pares.
- 7) Definir una lista que almacene los nombres de 4 personas. Contar cuántos de esos nombres tienen 5 o más caracteres.
- 8) Definir una lista vacía y luego solicitar la carga de 5 enteros por teclado y añadirlos a la lista. Imprimir la lista generada.
- 9) Realizar la carga de valores enteros por teclado, almacenarlos en una lista. Finalizar la carga de enteros al ingresar el cero. Mostrar finalmente todos los elementos cargados y el tamaño de la lista.
- 10) Almacenar en una lista los sueldos (valores float) de 7 operarios. Imprimir la lista y el promedio de sueldos.
- 11) Cargar por teclado y almacenar en una lista las alturas de 5 personas (valores float). Obtener el promedio de las mismas. Contar cuántas personas son más altas que el promedio.
- 12) Crear y cargar una lista con 5 enteros. Implementar un algoritmo que identifique el mayor valor de la lista.
- 13) Crear y cargar una lista con 5 enteros por teclado. Implementar un algoritmo que identifique el menor valor de la lista y su posición.
- 14) Ingresar por teclado los nombres de 5 personas y almacenarlos en una lista. Mostrar el nombre de la persona con el nombre más corto.
- 15) Cargar una lista con 5 elementos enteros. Imprimir el mayor y un mensaje si se repite dentro de la lista (es decir si dicho valor se encuentra en 2 o más posiciones en la lista)
- 16) Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas. Luego de realizar la carga por teclado de todos los datos imprimir los nombres de las personas mayores de edad (mayores o iguales a 18 años). Utilizar listas paralelas.

- 17) Crear y cargar dos listas con los nombres de 5 productos en una y sus respectivos precios en otra. Mostrar la cantidad de productos que tienen un precio mayor al primer producto ingresado.
- 18) Realizar un programa que pida la carga de dos listas numéricas enteras de 4 elementos cada una. Generar una tercer lista que surja de la suma de los elementos de la misma posición de cada lista. Mostrar esta tercer lista.
- 19) Realizar un programa que permita la registración de una cantidad determinada de alumnos y sus respectivas notas de exámenes y se deben procesar de acuerdo a lo siguiente:
- a) Ingresar la cantidad total de alumnos. (No se debe poder acceder a las opciones b,c y d si no se ingresó a la opción "a")
  - b) Ingresar nombre y nota de cada alumno (almacenar los datos en dos listas paralelas).
  - c) Realizar un listado que muestre los nombres, notas y condición del alumno. En la condición, colocar "Promocionado" si la nota es mayor o igual a 6, "Aprobado" si la nota es 4 o 5, y colocar "Reprobado" si la nota es inferior a 4.
  - d) Contar e imprimir por consola la cantidad de "Aprobados", "Promocionados" y "Reprobados".

## **Estructura FOR**

### **Actividades**

1. Mostrar los números ascendentes desde el 1 al 10
2. Mostrar los números descendentes desde el 10 al 1
3. Ingresar un número. Mostrar los números desde 0 hasta el número ingresado.
4. Ingresar un número y mostrar la tabla de multiplicar de ese número. Por ejemplo si se ingresa el numero 5:

$$5 \times 0 = 0$$

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15 \dots$$

5. Se ingresan un máximo de 10 números o hasta que el usuario ingrese el número 0. Mostrar la suma y el promedio de todos los números.
6. Imprimir los números múltiplos de 3 entre el 1 y el 10 (\*)
7. Mostrar los números pares que hay desde la unidad hasta el número 50 (\*)
8. Realizar un programa que permita mostrar una pirámide de números. Por ejemplo: si se ingresa el numero 5, la salida del programa será la siguiente:

```
1
12
123
1234
12345
```

9. Ingresar un número. Mostrar todos los divisores que hay desde el 1 hasta el número ingresado. Mostrar la cantidad de divisores encontrados.
10. Ingresar un número. Determinar si el número es primo o no.
11. Ingresar un número. Mostrar cada número primo que hay entre el 1 y el número ingresado. Informar cuántos números primos se encontraron.

## Ejercicios

Declarar una lista vacía, y manipularla mediante métodos de listas, mostrando los resultados después de cada operación:

- 1) Agregar 3 nombres de alumnos (ingresados por el usuario) al final de la lista.
- 2) Insertar un 4to nombre de alumno (ingresado por el usuario) al principio de la lista.
- 3) Extender la lista agregando 2 nombres adicionales al principio de la misma.
- 4) Contar e informar cuántas veces aparece un nombre ingresado por el usuario.
- 5) Eliminar de la lista un nombre ingresado por el usuario.
- 6) Eliminar el penúltimo nombre de la lista e informar que nombre se eliminó.
- 7) Ingresar un nombre por consola, buscarlo e informar su índice.
- 8) Ordenar los nombres descendientemente.
- 9) Crear una copia de la lista.
- 10) Limpiar la lista original eliminando todos sus elementos, mostrar la lista original y la copia.

## Ejercicios Diccionarios, Listas y ABM.

Desarrolle un programa con un menú que contenga los siguientes ítems y una opción para salir:

- 1) Alta: Desarrolle una función que permita cargar un diccionario a partir de los datos que ingresa el usuario por consola, los datos se componen por un nombre de alumno y 2 calificaciones (primer y segundo examen parcial).
- 2) Desarrolle una función que muestre una lista de alumnos y sus respectivos datos en filas y columnas, donde cada fila representa un alumno y cada columna representa uno de sus datos.
- 3) Modificación: Desarrolle una función que permita calcular el promedio de calificaciones a partir de una lista de alumnos. Recibe una lista de diccionarios por parámetro, calcula el promedio y lo agrega como un ítem más al diccionario.
- 4) Desarrolle una función que informe por cada alumno de la lista su estado académico (promedio de 1 al 4: desaprobado, 4 o 5: aprobado, y 6 o más: promocionado).
- 5) Desarrolle una función que informe las notas y el promedio del alumno cuyo nombre recibe por parámetro, en caso de no encontrarlo deberá imprimir un mensaje de error.
- 6) Baja Física: Desarrolle una función que pueda eliminar a un alumno de la lista de alumnos. El alumno a eliminar deberá seleccionarlo el usuario por terminal, validar que exista antes de eliminarlo, y en caso de que no exista mostrar un mensaje de error.
- 7) Baja Lógica: Desarrolle una función que pueda dar de baja lógicamente a un alumno de la lista. Deberá recibir por parámetro el nombre del alumno a eliminar y agregarle un estado (bool) activo o inactivo. Modificar la función que muestra los alumnos haciendo que ignore a todos los alumnos inactivos.
- 8) Desarrolle una función que busque al alumno con el mejor o con el peor promedio. Informar sus nombres y sus respectivos promedios.
- 9) Desarrolle una función que ordene a los alumnos por promedio ASC/DESC.
- 10) Desarrolle una función que calcule e informe la cantidad de alumnos según su estado académico (desaprobado, aprobado o promocionado).

## Ejercicios prácticos.

- 1) Desarrollar una función que reciba como parámetros fecha actual y fecha de nacimiento; y retorne la edad.
- 2) Desarrollar una función que reciba una patente que tendrá tres letras y tres números o tres números y tres letras. Deberá retornar auto o moto, si la patente es tres letras y tres números o tres números y tres letras respectivamente.
- 3) Desarrollar una función que reciba como parámetros número de DNI y [MASCULINO|FEMENINO|EMPRESA]. Deberá retornar el CUIL/CUIT formado por:

CUIL/T: Son 11 números en total:

XY – 12345678 – Z

XY: Indican el tipo (Masculino, Femenino o una empresa)

12345678: Número de DNI

Z: Código Verificador

Algoritmo: Se determina XY con las siguientes reglas:

Masculino: 20

Femenio: 27

Empresa: 30

Se multiplica XY 12345678 por un número de forma separada:

$X * 5$

$Y * 4$

$1 * 3$

$2 * 2$

$3 * 7$

$4 * 6$

$5 * 5$

$6 * 4$

$7 * 3$

$8 * 2$

Se suman dichos resultados. El resultado obtenido se divide por 11. De esa división se obtiene un Resto que determina Z.

Si el resto es 0 = Entonces  $Z = 0$ .

Si el resto es 1 = Entonces se aplica la siguiente regla:

\*Si es hombre:  $Z = 9$  y XY pasa a ser 23

\*Si es mujer:  $Z = 4$  y XY pasa a ser 23

Caso contrario Z pasa a ser  $(11 - \text{Resto})$ .

*Ejemplo de Cálculo de CUIL CUIT:*

Masculino DNI 12345678

1-Determinar el Tipo

XY es 20

Hacemos el cálculo

$$2 * 5 = 10$$

$$0 * 4 = 0$$

$$1 * 3 = 3$$

$$2 * 2 = 4$$

$$3 * 7 = 21$$

$$4 * 6 = 24$$

$$5 * 5 = 25$$

$$6 * 4 = 24$$

$$7 * 3 = 21$$

$$8 * 2 = 16$$

Realizamos la suma de  $(10+0+3+4+21+24+25+24+21+16) = 148$

Dividimos por 11 para obtener Z (el código verificador)

$$148 / 11 = 13,4545 \rightarrow 13 \text{ (Redondeo)}$$

Obtenemos el resto de la división:

$$148 - (13 * 11) = 5$$

Determinamos Z:

$$11 - 5 = 6$$

**Conclusión: CUIL-CUIT 20-12345678-6**

## Ejercicios.

- 1) Desarrollar una función que determine si un número entero es par o impar. Debe recibir un número por parámetro y devolver True en caso de que sea par, de lo contrario devolverá False.
- 2) Desarrollar una función que reciba un carácter y determine si ese carácter está comprendido entre a...z o A...Z, en caso afirmativo devolverá True, de lo contrario False.
- 3) Desarrollar una función que reciba un carácter y determine si ese carácter está comprendido entre 0...9, devolverá un valor boolean indicando si el carácter recibido es numérico o no.
- 4) Desarrollar una función que permita validar un número entero. Deberá recibir por parámetro el número a validar, y dos números que representan el rango mínimo y máximo permitido. Devolverá True en caso de ser válido, False de lo contrario.
- 5) Desarrollar una función que se encargue de solicitar un número entero al usuario, validarlo (reutilizando la función del punto anterior) y retornar el número validado y transformado a entero. Deberá recibir por parámetro un mensaje que se le mostrará al usuario y los rangos de validación.
- 6) Desarrollar una función que se encargue de solicitar una cadena de caracteres al usuario, validarla y retornar la misma. Deberá recibir como parámetro un mensaje para indicarle al usuario y 1, 2 o 3 cadenas de caracteres que representarán las opciones válidas de ingreso.
- 7) Desarrollar una función que se encargue de medir el largo de una cadena de caracteres, deberá recibir por parámetro la cadena de caracteres a evaluar y devolverá un número entero representando la longitud de la cadena recibida.
- 8) Desarrollar una función que determine si un número que recibirá por parámetro es primo. Si es primo deberá devolver un True, de lo contrario False.
- 9) Desarrollar una función que recibirá un número entero por parámetro, y devolverá el resultado del factorial de ese número.
- 10) Desarrollar una función que verifique el DNI de una persona, la misma recibirá una cadena de caracteres (se asume que solamente contiene caracteres numéricos). Si la cantidad de caracteres es menor a 6 o mayor a 8, retornara False. Si la cantidad de caracteres está comprendida entre 6 y 8 devolverá True.
- 11) Desarrollar una función que complete el número de DNI de una persona. Recibirá una cadena de caracteres (se asume que solamente contiene caracteres numéricos), deberá medirla y completar con ceros a la izquierda hasta llegar a un total de 8 caracteres, y devolviendo la cadena resultante. Ej: Se ingresa "123456", y devolverá "00123456".
- 12) Desarrollar una función que transforme una cadena de caracteres numérica a su equivalente en letras. Recibirá por parámetro la cadena a transformar y devolverá el resultado en letras. Ej: "987" -> "novecientos ochenta y siete". Como máximo tomar el número más grande de 3 dígitos.

**Nota: Incluir documentación y tipos de datos para los parámetros y retorno para todas las funciones.**

**Nota2: Opcionalmente pueden agregar validaciones de TIPOS de datos como: `type(dato) == str`**

**Y en caso de no coincidir con el tipo de dato que espera recibir la función, devolver un None.**



## Matrices.

### Ejercicios.

- 1- Crear y cargar una matriz de 4x4 con datos numéricos aleatorios entre 1 y 9 inclusive, en formato string. Mostrar los números pares de esa matriz (en filas y columnas) y reemplazar los números impares con cadenas vacías para ocultarlos.
- 2- Generar una matriz de 3x3 cargando datos numéricos del 1 al 9 inclusive en celdas aleatorias, sin que se repitan los números (al estilo Sudoku).
- 3- Desarrollar una función que reciba 2 matrices, los sume y devuelva la matriz resultante, sin modificar las matrices originales.
- 4- Desarrollar una función que reciba una matriz y un número escalar, multiplicar la matriz por el número escalar y retornar la matriz resultante, sin modificar la matriz original.
- 5- Desarrollar una función que reciba dos matrices, y las multiplique entre sí. Se debe validar que las matrices recibidas por parámetro se puedan multiplicar entre sí y devolver la matriz resultante, sin alterar las matrices originales, caso contrario la función devolverá un None.
- 6- Desarrollar un programa que cuente con un menú y las siguientes opciones:
  - a) Generar una matriz con números aleatorios. Las dimensiones y los rangos de números se deben pasar por parámetros de la función generadora. No se debe poder acceder a las demás opciones si la matriz no fue generada.
  - b) Mostrar la matriz generada.
  - c) Determinar si la matriz contiene series de números consecutivos (en horizontal o en vertical).
  - d) Determinar la cantidad total de series (las series de números consecutivos de más de dos números cuentan como una sola).
  - e) Determinar el largo de la serie más corta, y mostrar todas las series de ese largo.
  - f) Determinar el largo de la serie más larga, y mostrar todas las series de ese largo.
  - g) Salir.

## Ejercicios

1) Agregar las siguientes funciones al menú del trabajo práctico sobre diccionarios:

- Desarrolle una función que permite guardar una lista de diccionarios en formato JSON (utilizar esta función como opción 11 y añadir guardado automático a la opción de salir, con confirmación por parte del usuario).
- Desarrolle una función que permite leer una lista de diccionarios desde un archivo JSON (se carga automáticamente al correr el programa).

2) Desarrolle y pruebe las siguientes funciones en un archivo separado al de trabajo práctico sobre diccionarios:

- Desarrolle una función que genere una matriz de 9 filas por 9 columnas con números aleatorios del 1 al 9 inclusive.
- Desarrolle una función que guarde la matriz generada aleatoriamente en un archivo con formato CSV, donde las columnas están separadas por comas y las filas por saltos de línea (\n).
- Desarrolle una función para poder leer un archivo CSV y a partir de su contenido construir una matriz en Python.
- Desarrolle una función que muestre la matriz construida a partir de la lectura de un archivo CSV dándole formato de matriz en la terminal.
- Desarrolle una función que reciba una matriz de números enteros y guarde todos los números pares en un archivo de texto con el nombre "numeros\_pares.txt" y todos los números impares en otro archivo con el nombre "numeros\_impares.txt".
- Desarrolle una función que pueda agregar su nombre, apellido y división al final del archivo que guarda la matriz aleatoria.

## Sucesión de Fibonacci.

1) Desarrollar función `calcular_fibonacci`.

Parámetros: La misma recibirá un número entero (int) mayor o igual a cero (0).

Funcionalidad: La función deberá calcular el número  $n$ -ésimo en la sucesión de Fibonacci.

Si  $n = 0$ , deberá devolver 0.

Si  $n = 1$ , deberá devolver 1.

Para cualquier  $n > 1$ , el resultado será la suma de los dos números anteriores de la secuencia.

Retorno: El resultado calculado previamente.

Por ejemplo:

- $f_0 = 0$
- $f_1 = 1$
- $f_2 = 1$
- $f_3 = 2$
- $f_4 = 3$
- $f_5 = 5$
- $f_6 = 8$
- 

2) Desarrollar función `mostrar_serie_fibonacci`.

Parámetros: La misma recibirá un número entero (int) mayor o igual a cero (0).

Funcionalidad: La función deberá mostrar la secuencia completa hasta el número  $n$ -ésimo en la sucesión de Fibonacci, incluyendo a este último.

Retorno: None

## Suma de números consecutivos.

Por ejemplo:

$$1 + 2 = 3$$

$$2 + 3 = 5$$

1) Desarrollar función `determinar_suma_consecutiva`.

Parámetros: La misma recibirá un número entero (int), y cualquier otro parámetro que considere necesario.

Funcionalidad: Deberá determinar si el número ingresado como parámetro se puede obtener con la suma de dos (2) números consecutivos anteriores.

Retorno:

True si el número se puede obtener de la suma de números consecutivos anteriores.

False si el número NO se obtiene de la suma de números consecutivos anteriores.

## Ejercicio Integrador 01

La división de higiene está trabajando en un control de stock para productos sanitarios. Debemos realizar la carga de 5 (cinco) productos de prevención de contagio, de cada una debe obtener los siguientes datos:

1. El tipo (validar "barbijo", "jabón" o "alcohol")
2. El precio: (validar entre 100 y 300)
3. La cantidad de unidades ( no puede ser 0 ni negativo y no debe superar las 1000 unidades)
4. La marca y el Fabricante.

Se debe informar lo siguiente:

- A. Del más caro de los barbijos, la cantidad de unidades y el fabricante.
- B. Del ítem con más unidades, el fabricante.
- C. Cuántas unidades de jabones hay en total.

Ejemplo de mensaje de consola:

```
El mas caro de los barbijos tiene una cantidad de: 100 unidades y es fabricado por: L1
El item con mas unidades es: L1 y es: 100
La cantidad de jabones es: 320
Gracias por usar el programa
```