

Trabajo Práctico 6: Funciones

Objetivos:

- Comprender el concepto de modularización a través de subprogramas.
- Construir programas a partir de módulos reutilizables.

Codificar funciones apropiadas para cumplir con los siguientes objetivos, escribiendo también el programa principal correspondiente:

Ejercicio 1: Escribir una función que reciba como parámetros dos números enteros. Calcular y devolver el resultado de la multiplicación de ambos valores utilizando solamente sumas. Por ejemplo $4 * 3 = 4 + 4 + 4$.

Ejercicio 2: Dados dos parámetros enteros A y B, obtener A^B (A elevado a la B) mediante multiplicaciones sucesivas, utilizando la función del ejercicio anterior para multiplicar. Por ejemplo $4^3 = 4 * 4 * 4$.

Ejercicio 3: Imprimir una columna de asteriscos, donde su altura se recibe como parámetro.

Ejercicio 4: Devolver True si el número entero recibido como primer parámetro es múltiplo del segundo, o False en caso contrario. Ejemplo: `esmultiplo(40, 8)` devuelve True y `esmultiplo(50, 3)` devuelve False.

Ejercicio 5: Desarrollar la función `signo(n)`, que reciba un número entero y devuelva 1, -1 o 0 según el valor recibido sea positivo, negativo o nulo.

Ejercicio 6: Escribir la función `comparar(a, b)` que reciba como parámetros dos números enteros y devuelva 1 si el primero es mayor que el segundo, 0 si son iguales o -1 si el primero es menor que el segundo. En este ejercicio debe aprovecharse la función del ejercicio anterior. Ejemplo: `comparar(4, 2)` devuelve 1, y `comparar(2, 4)` devuelve -1.

Ejercicio 7: Calcular y devolver el Máximo Común Divisor de dos enteros no negativos, basándose en las siguientes fórmulas matemáticas:

- $MCD(X, X) = X$
- $MCD(X, Y) = MCD(Y, X)$
- Si $X > Y \Rightarrow MCD(X, Y) = MCD(X - Y, Y)$.

Ejemplo: `MCD(40, 15)` devuelve 5.

Ejercicio 8: La raíz cuadrada de un número positivo n puede calcularse mediante la siguiente fórmula de Newton:

$$\sqrt{n} \approx \frac{\frac{n}{a} + a}{2}$$

donde a es una aproximación a n . Al aplicar repetidamente esta fórmula reemplazando a por la aproximación obtenida en el paso anterior, se obtiene cada vez una aproximación mejor. Desarrollar un programa que calcule la raíz cuadrada aproximada de un número entero positivo n utilizando como primera aproximación a $n/2$. Detener el proceso cuando la diferencia entre dos cálculos sucesivos sea menor a 0,0001.

Ejercicio 9: Escribir una función que reciba como parámetros un número de día, un número de mes y un número de año y devuelva la cantidad de días que faltan hasta fin de mes. Luego desarrollar tres programas para:

- Ingresar una fecha formada por tres enteros (día, mes y año) e imprimir la cantidad de días que faltan hasta fin de año.
- Ingresar una fecha formada por tres enteros (día, mes y año) e imprimir la cantidad de días transcurridos en ese año hasta esa fecha.
- Ingresar dos fechas formadas por tres enteros (día, mes y año) e imprimir cuánto tiempo transcurrió entre ambas, expresando el resultado en años, meses y días.

Los tres programas deben realizar su trabajo a través de la función indicada al comienzo.

Ejercicio 10: Extraer un dígito de un número. La función recibe como parámetros dos números enteros, uno será del que se extraiga el dígito y el otro indica qué cifra se desea obtener. La cifra de la derecha se considera la número 0. Retornar el valor -1 si no existe el dígito solicitado. Tener en cuenta que el número puede ser positivo o negativo. Ejemplo: `extraerdigito(12345, 1)` devuelve 4, y `extraerdigito(12345, 8)` devuelve -1.

Ejercicio 11: Obtener el dígito central de un número entero pasado como parámetro, sólo si la cantidad de dígitos es impar. Si la longitud fuera par devolver -1. Ejemplo: `digitocentral(12345)` devuelve 3, y `digitocentral(123456)` devuelve -1.

Trabajo Práctico 7: Estructuras de Datos – Listas

Objetivos:

- Introducir el concepto de estructuras de datos.
- Familiarizarse con el uso de Listas en Python, conocidas como arreglos o vectores en otros lenguajes de programación.

Desarrollar los siguientes programas o funciones, según corresponda:

Ejercicio 1: Escribir una función para ingresar desde el teclado una serie de números entre A y B y guardarlos en una lista. En caso de ingresar un valor fuera de rango la función mostrará un mensaje de error y solicitará un nuevo número. Para finalizar la carga se deberá ingresar -1. La función recibe como parámetros los valores de A y B, y devuelve la lista cargada (o vacía, si el usuario no ingresó nada) como valor de retorno. Tener en cuenta que A puede ser mayor, menor o igual a B.

En los ejercicios 2 a 6 utilizar la función del ejercicio 1 para ingresar datos en una lista y:

Ejercicio 2: Calcular la suma de los números de la lista.

Ejercicio 3: Determinar si la lista es capicúa (palíndromo). Una lista capicúa se lee de igual modo de izquierda a derecha y de derecha a izquierda. Por ejemplo, [2, 7, 7, 2] es capicúa, mientras que [2, 7, 5, 2] no lo es.

Ejercicio 4: Escribir una función para contar cuántas veces aparece un valor dentro de la lista. La función recibe como parámetros la lista y el valor a buscar, y devuelve un número entero.

Ejercicio 5: Desarrollar una función que reciba la lista como parámetro y devuelva una nueva lista con los mismos elementos de la primera, pero en orden inverso. Por ejemplo, si la función recibe [5, 7, 1] debe devolver [1, 7, 5].

Ejercicio 6: Escribir una función para devolver una lista con todas las posiciones que ocupa un valor pasado como parámetro, utilizando búsqueda secuencial en una lista desordenada. La función debe devolver una lista vacía si el elemento no se encuentra en la lista original.

Ejercicio 7: Ídem anterior, utilizando búsqueda binaria sobre una lista ordenada.

Ejercicio 8: Rellenar una lista con números enteros entre 0 y 100 obtenidos al azar e imprimir el valor mínimo y el lugar que ocupa. Tener en cuenta que el mínimo puede estar repetido, en cuyo caso deberán mostrarse todas las posiciones en las que se encuentre. La carga de datos termina cuando se obtenga un 0 como número al azar, el que no deberá cargarse en la lista.

Ejercicio 9: Crear una lista de N números generados al azar entre 0 y 100 pero sin elementos repetidos. El valor de N se ingresa por teclado. Resolver este problema utilizando dos estrategias distintas:

- Impidiendo el agregado de elementos repetidos
- Eliminando los duplicados luego de generar la lista. Asegurarse que la cantidad final de elementos sea la solicitada.

- Ejercicio 10:** Eliminar de una lista de números enteros los valores que se encuentren en una segunda lista. Imprimir la lista original, la lista de valores a eliminar y la lista resultante. Ambas listas deben rellenarse con números al azar. La cantidad y el rango de los valores los decide el programador.
- Ejercicio 11:** Cargar dos listas de números A y B con N números al azar entre 1 y 100, donde N se ingresa por teclado. Mostrar ambas listas por pantalla. Luego construir e imprimir tres nuevas listas C, D y E que contengan:
- La concatenación de los valores pares de A con los impares de B. (valores pares o valores impares se refiere a los elementos propiamente dichos y no a sus posiciones).
 - La concatenación de los valores impares de A con el reverso de los valores pares de B.
 - La intercalación de los elementos de A y B.
- Ejercicio 12:** Dada una lista ordenada de números llamada A y un nuevo número N, desarrollar un programa que agregue el elemento N dentro de la lista A, respetando el ordenamiento existente. El programa deberá detectar automáticamente si el ordenamiento es ascendente o descendente antes de realizar la inserción. No se permite añadir el elemento al final y reordenar la lista.
- Ejercicio 13:** Leer dos listas de números M y N y ordenarlas de menor a mayor. Luego se solicita generar e imprimir una tercera lista que resulte de intercalar los elementos de M y N. La nueva lista también debe quedar ordenada, sin utilizar ningún método de ordenamiento en ella.
- Ejercicio 14:** Una escuela necesita conocer cuántos alumnos cumplen años en cada mes del año, con el propósito de ofrecerles un agasajo especial en su día. Desarrollar un programa que lea el número de legajo y fecha de nacimiento (día, mes y año) de cada uno de los alumnos que concurren a dicha escuela. La carga finaliza con un número de legajo igual a -1. Emitir un informe donde aparezca -mes por mes- cuántos alumnos cumplen años a lo largo del año. Imprimir también una leyenda que indique cuál es el mes con mayor cantidad de cumpleaños.