

Universidad Nacional del Litoral  
**Facultad de Ingeniería y Ciencias Hídricas**  
Departamento de Informática



Ingeniería Informática

---

## **FUNDAMENTOS DE PROGRAMACIÓN**

### **UNIDAD 8** **Funciones**

Guía de trabajos prácticos  
2023

## UNIDAD 8

# Funciones

---

### Ejercicio 8.1

Escriba prototipos que cumplan con las características de las siguientes funciones propuestas. Proponga los elementos que considere necesarios.

- Una función con nombre *facto()* que sirva para calcular el factorial de un número entero.
- Una función *hipot()* que retorne la hipotenusa de un triángulo rectángulo.
- Una función llamada *es\_primo* que permita determinar si un número es primo.
- Una función para mostrar un arreglo\* en pantalla.
- Una función con nombre *division\_con\_resto()* que devuelva el cociente entero y el resto de la división entre 2 enteros.
- Una función llamada *intercambia()* que permita intercambiar los valores de dos variables de tipo caracter.
- Una función para insertar un elemento en un arreglo\*.

*\*Nota: El tipo de dato para decir “arreglo 1D de enteros” en C++ es “vector<int>”. Lo aprenderemos a utilizar en la próxima unidad. Por ahora solo debe declarar el prototipo, no implementar la función completa.*

### Ejercicio 8.2

Reescriba el ejercicio 7.12 (determinar e informar los 1000 primeros números primos) utilizando una función auxiliar para determinar si un número es primo.

### Ejercicio 8.3

Escriba una función que permita calcular la diferencia en años entre dos fechas con formato *aaaammdd* y utilícela desde un programa cliente.

### Ejercicio 8.4

Escriba una función que determine el mayor de 3 valores enteros positivos. Modifique la función para que, utilizando parámetros por defecto, sirva también para determinar el mayor de 2 parámetros.

### Ejercicio 8.5

Desarrolle una función llamada *Intercambia(...)* que reciba dos variables enteras por referencia e intercambie sus valores. Escriba dos sobrecargas de la función que permitan realizar el mismo procedimiento con dos datos de tipo real, y dos de tipo carácter. Pruebe las funciones desde un programa cliente.

### Ejercicio 8.6

Escriba un programa que permita calcular las raíces de una ecuación cuadrática de la forma  $ax^2+bx+c = 0$  conociendo los valores de sus coeficientes. Para ello:

- a. Implemente una función para determinar si las raíces serán reales o complejas.
- a. Implemente una función para calcular las raíces en el caso de que sean reales.
- a. Implemente una función que calcule los componentes del complejo (parte real y parte imaginaria) en caso de que las raíces sean complejas conjugadas.

Proponga para las funciones los parámetros y valores de retorno que considere apropiados. El programa debe informar el tipo y valor de las raíces.

### Ejercicio 8.7

- a) En cierta materia, un alumno es regular si tiene al menos 50% en cada parcial (o en su recuperatorio, queda siempre la mejor nota) y 60% de asistencia. Y es promocionado si tiene al menos 80% en cada parcial y 80% de asistencia. Escriba una función que reciba los datos de un alumno y retorne su condición.
- b) Desarrolle un programa cliente que permita ingresar los nombres, el porcentaje de asistencia, y las 4 calificaciones (parcial 1, parcial 2, recuperatorio 1, y recuperatorio 2) de cada uno de los  $N$  alumnos del curso. El programa debe, haciendo uso de la función, contar y mostrar cuántos alumnos quedan en condición de regular y cuántos de promoción.

### Ejercicio 8.12

El [algoritmo de Euclides](#) es un método antiguo y eficaz para calcular el máximo común divisor (MCD) de dos números enteros. Fue propuesto originalmente por Euclides en su obra *Elementos* y enuncia que el MCD entre dos números  $a$  y  $b$  puede obtenerse de manera recursiva aplicando las siguientes propiedades:

- Si  $a > b$ , entonces  $\text{MCD}(a,b) = \text{MCD}(a-b,b)$
- Si  $a < b$ , entonces  $\text{MCD}(a,b) = \text{MCD}(a, b-a)$
- Si  $a = b$ , entonces  $\text{MCD}(a,b) = a$



## Cuestionario

---

1. Mencione las ventajas del empleo de subprogramas.
2. ¿Qué es el prototipo de una función C++? ¿Dónde debe plantearse?
3. A continuación se propone un alias o referencia a una posición de memoria que no ha sido identificada antes con una variable: `int &x=3`. ¿Acepta el compilador C++ esta proposición? Explique.
4. ¿Es posible que una función no devuelva resultados? ¿Cómo reconoce una función C++ que no retorna un resultado?
5. Defina: a) argumentos o parámetros formales; b) parámetros actuales.
6. Explique pasaje de parámetros por valor y pasaje por referencia.
7. ¿Pueden emplearse nombres de argumentos en una función C++ que sean coincidentes con nombres de otras variables declaradas en la función main?
8. ¿Puede una función utilizar variables declaradas en el programa cliente?
9. Describa las posibilidades de empleo de la sentencia return.
10. ¿Es posible llamar a una función con menos parámetros actuales que los formales propuestos en el prototipo? Explique.
11. ¿Por qué no es conveniente realizar acciones de entrada / salida en una función?
12. ¿Cuáles son las condiciones para que una función sea recursiva?
13. ¿Cuáles son las ventajas y desventajas de utilizar funciones recursivas?
14. ¿Es obligatorio colocar identificadores en los parámetros formales del prototipo de una función C++?
15. ¿Qué entiende por sobrecarga de funciones? ¿Cómo detecta el compilador cuál función debe ejecutar ante una llamada en el caso de sobrecarga?

## Ejercicios Adicionales

---

### Ejercicio 8.101

Un año "bisiesto" es un año en el que el mes de Febrero tiene un día más (29 en lugar de 28). Un año es bisiesto si es divisible por 4, a menos que sea divisible por 100 y no por 400. Ejemplos:

- 2004 es porque es divisible por 4.
- 2005 no es porque no es divisible por 4.
- 2100 es divisible por 4, pero no es porque es divisible por 100 pero no por 400.
- 2000 sí porque, aunque es divisible por 100, también es divisible por 400.

**a)** Escriba una función para determinar si un año es bisiesto.

**b)** Escriba otra función que reciba una fecha en formato "AAAAMMDD" (Ej: 20240513 sería el 13 de Mayo de 2024) y verifique si es correcta:

- El mes debe estar entre 1 y 12.
- El día debe estar entre 1 y la cantidad de días del mes (considerando si es o no un año bisiesto para el caso de febrero).

Diseñe la función de forma tal que además de retornar si es o no correcta, en caso de no serlo permita saber dónde está el error (si en el día, en el mes, o en ambos).

**c)** Utilice esta función en un programa cliente que permita ingresar la fecha actual y la fecha de nacimiento de una persona para luego calcular e informar la edad de la persona. El programa debe validar en cada caso que la fecha ingresada sea correcta, y además que la actual sea mayor a la de nacimiento (es decir, si no es correcta, informar el error y solicitarla nuevamente).

### Ejercicio 8.102

Si se tienen dos tiempos (horas, minutos y segundos de cada uno) y se intentan sumar/restar simplemente sumando horas con horas, minutos con minutos y segundos con segundos. El resultado puede ser un tiempo incorrecto. Ejemplos:

\*  $1:59:00 + 1:03:05 = 2:62:05$  (cuando debería ser  $3:02:05$ )

\*  $1:03:05 - 0:05:10 = 1:-02:-05$  (cuando debería ser  $0:57:55$ )

**a)** Escriba una función que dado uno de estos resultados incorrectos lo corrija.

**b)** Escriba dos funciones, una para sumar y otra para restar mediciones de tiempos correctamente.

**c)** Escriba un programa cliente para verificar el correcto funcionamiento de las funciones.

### Ejercicio 8.103

**a)** Escriba una función que reciba el módulo y el ángulo de un vector 2D, y retorne los componentes x e y. **b)** Escriba otra función que reciba las componentes x e y, y retorne el módulo y el ángulo.

**c)** Todo cálculo con reales en la PC puede tener errores por falta de precisión. Escriba una función auxiliar para comparar dos reales buscando que sean muy muy parecidos (diferencia menor a  $1 \times 10^{-5}$  si es *float*,  $1 \times 10^{-12}$  si es *double*) y no exactamente iguales.

**d)** Escriba un programa cliente para verificar automáticamente las dos primeras

funciones (que si le da el resultado de la 1ra función a la 2da, se obtenga nuevamente la entrada, y lo mismo invirtiendo las funciones) generando valores aleatorios y comparando los resultados con la 3ra.

### Ejemplos de Recursión

#### Ejercicio 8.8

Escriba una función recursiva que permita calcular la potencia de un número elevado a un exponente entero y utilícela desde un programa cliente. *Ayuda: recuerde que  $a^n = a * a^{n-1}$ .*

#### Ejercicio 8.9

Escriba una función recursiva que permita calcular la potencia de un número elevado a un exponente entero, sabiendo que:

- $a^n = a^{n/2} * a^{n/2}$ , si n es par y mayor que cero
- $a^n = a * a^{n-1}$ , si n es impar y mayor que cero

Compare esta solución -en cuanto a la cantidad de llamadas que realiza-, contra la versión anterior (por ejemplo para el caso  $2^{30}$ ).

#### Ejercicio 8.10

La función  $H(x)$  se define de la siguiente forma:

- $4 * x$  si  $x < 4$
- $3 * H(x - 2) + 1$  si  $x \geq 4$

Escriba una función recursiva que calcule el valor de la función y obtenga, con un programa cliente, los valores de  $H(7)$ ,  $H(5)$  y  $H(12)$ .

#### Ejercicio 8.11

Los valores 0, 1, 1, 2, 3, 5, 8, 13 corresponden a los ocho primeros términos de la célebre [sucesión de Fibonacci](#). En dicha sucesión, el i-ésimo valor ( $t_i$ ) se calcula como:  $t_i = t_{i-1} + t_{i-2}$ . Es decir, cada término se obtiene de la suma de los dos términos anteriores, a excepción del primero y el segundo, los cuales son, por definición, iguales a 0 y 1 respectivamente.

- a) Escriba una función recursiva que permita calcular el i-ésimo término de la sucesión de Fibonacci, según la definición anterior.
- b) Luego utilícela para mostrar los 20 primeros términos de la misma.
- c) Analice la cantidad de llamadas que se ejecutan para obtener  $t_3$ ,  $t_4$  y  $t_5$ .

