

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  
2016

## 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 llamada *division\_entera()* que devuelva el cociente entero de la división entre 2 enteros.
- 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 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 `character`.

### 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 `character`. 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:

- Implemente una función para calcular el determinante.
- Implemente una función para calcular las raíces en el caso de que sean reales (la función debe, a su vez, utilizar la función implementada en a).
- Implemente una función que calcule los coeficientes 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

Se ingresan los nombres y edades de los 457 miembros de una asociación cooperadora. Determinar el nombre y la edad del mayor de los integrantes. Utilizar una función llamada *validar\_edad()* para validar la entrada de los datos correspondientes a las edades, los cuales deben estar dentro el rango 12...90. La función debe indicar un mensaje de error y permitir el reingreso del dato mientras este no se encuentre en el rango indicado y devolver la primer edad válida que el usuario ingrese.

### 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. Luego utilícela para mostrar los 20 primeros términos de la misma.

b) Evalúe la cantidad de llamadas que se ejecutan para  $\text{Fibonacci}(5)$ . Es posible codificar una función recursiva para calcular el  $n$ -ésimo término de Fibonacci, de forma tal que la recursividad sea lineal? (en el código de la función Fibonacci, sólo debe haber una llamada a Fibonacci, en lugar de dos, como en el ítem a). Si es así, codifíquela!

### 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  $\text{MCD}(a, b) = \text{MCD}(b, a) = a = b$

## 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.A

Escriba dos funciones no recursivas que retornen el mismo resultado que las funciones recursivas presentadas a continuación:

```
int f1(int a,int b) {  
    if (a==0) return 0;  
    return b+f1(a-1,b);  
}
```

```
int f2(int a,int b) {  
    if (a<b) return 0;  
    return 1+f2(a-b,b);  
}
```

¿En qué casos fallan estas funciones recursivas?