

# Ejercicios con Funciones

Programación en C

# Programa 1

---

Realice un programa `cuadrados.c` en lenguaje C que lea un número entero  $n$  en la función principal y se lo pase como argumento a la función *cuadrados* para calcular la suma de los cuadrados de los primeros  $n$  números enteros.

La función *cuadrados* devolverá:

- Si el número es negativo o 0  $\rightarrow$  -1.
- Si el número es positivo  $\rightarrow$  la suma de los cuadrados de los primeros  $n$  números enteros.

La función *main* imprimirá el resultado devuelto y si es -1, un mensaje de error.



# Programa 2

---

Realice un programa *divisores.c* en lenguaje C que tenga una función *calculadivisores*.

La función *calculadivisores*:

- se le pasa un número entero y devuelve el número de divisores de dicho número.
- imprimirá por pantalla cada divisor.

La función *main*:

- leerá un número entero por teclado
- llamará a la función *calculadivisores*
- imprimirá el número de divisores del número en cuestión.



# Programa 3

---

Realice un programa *pitagoras.c* en lenguaje C que tenga una función *pitagoras*.

La función *pitagoras*:

- recibe como argumentos dos números (correspondientes a los catetos)
- calculará la hipotenusa mediante el Teorema de Pitágoras y devolverá dicho resultado.

La función principal (*main*):

- Pedirá al usuario el valor de los catetos.
- Llamará a la función *pitagoras*.
- Imprimirá el resultado devuelto por la función por pantalla.

Nota: usad la función *pow* de la librería matemática (*math.h*). Para usar esta librería es necesario compilar con la opción *-lm*.

```
$> gcc -o distancia distancia.c -lm
```

---



# Programa 4

---

Realice un programa `distancia.c` en lenguaje C que contenga la función *distancia*.

La función *main*:

- leerá 6 números reales correspondientes con las coordenadas de 2 puntos en el espacio  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ .
- llamará a la función *distancia* para que calcule la distancia euclídea entre dichos puntos.
- imprimirá el resultado.

La función *distancia*:

- recibirá 6 números reales correspondientes con las coordenadas de los 2 puntos.
- Calculará y devolverá la distancia euclídea entre dichos puntos.

La distancia euclídea es:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

**Nota:** usad las funciones *pow* y *sqrt* de la librería matemática (*math.h*).

---



# Programa 5

---

Realice un programa en lenguaje C que contenga la función *esprimo*.

La función *main*:

- lee un entero
- llamará a la función *esprimo*
- imprime un mensaje indicando si el número es o no primo en función del resultado de la función.

La función *esprimo*:

- recibe como argumento el entero leído.
- devuelve 1 si el número es primo y 0 si el número no es primo.



# Programa 6

---

Realice un programa *potencia.c* en lenguaje C que lea dos números naturales en la función principal y se los pase como argumentos a la función *potencia*.

La función *potencia*:

- recibe 2 números enteros (base y exponente)
- calcula la *base* elevado al *exponente* (sin usar la función *pow*) y devuelve el resultado.

La función *main*:

- imprimirá el resultado devuelto o un mensaje de error.



# Programa 7

---

Realice un programa *ohm.c* en lenguaje C que implemente la Ley de Ohm ( $V=R*I$ ).

La función *main*:

- presentará un menú para saber cuál es la incógnita (V, R, I).
- llamará a la función *ley\_de\_ohm* pasándole un valor que indique cuál es la incógnita.
- imprimirá el resultado que le devuelva la función o un mensaje de error si la opción elegida en el menú no es correcta.

La función *ley\_de\_ohm*:

- leerá los otros dos parámetros que falten.
- calculará y devolverá el resultado a la función *main*.

