

1. Escribe una función que tome 3 parámetros: dos de tipo entero y uno de tipo carácter. La función deberá sumar, restar, multiplicar o dividir los valores de los dos primeros parámetros dependiendo de la operación indicada en el tercer parámetro, y devolver el resultado.
2. Sobrecarga la función del ejercicio anterior para que se pueda operar con enteros y con decimales. Haz un programa que utilice las dos funciones, con enteros y con decimales.
3. Se dice que un número entero es primo si sólo es divisible entre 1 y entre sí mismo. Por ejemplo, 2, 3, 5 y 7 son primos, pero 4, 6, 8 y 9 no lo son.
 - a) Escribe una función que determine si un número es primo o no.
 - b) Utiliza esta función en un programa que muestre todos los números primos entre 1 y 10.000.
4. Escribe un programa que lea un número entero y lo descomponga en factores primos. Utiliza la función realizada en el ejercicio anterior.
Ejemplo: $18 = 2 * 3 * 3$
 $11 = 11$
 $35 = 5 * 7$
 $40 = 2 * 2 * 2 * 5$
5. Se dice que un número entero es un número perfecto si la suma de sus divisores propios (incluyendo el 1 y sin incluirse él mismo) da como resultado el mismo número. Así, 6 es un número perfecto, porque sus divisores propios son 1, 2 y 3; y $6 = 1 + 2 + 3$. Los siguientes números perfectos son 28, 496 y 8128.
 - a) Escribe una función llamada **perfecto** que determine si el parámetro es perfecto o no.
 - b) Realiza otra función que dado un número perfecto, imprima los factores para confirmar que el número es perfecto. Si no lo es, que no haga nada.
 - c) Utiliza estas funciones en un programa que muestre todos los números perfectos entre 1 y 10.000 con sus correspondientes factores.
6. El máximo común divisor (MCD) de dos enteros es el entero más grande que es divisor exacto de los dos números. Escribe una función que devuelva el máximo común divisor de dos enteros.

RECURSIVIDAD

7. Escribe un programa que calcule el factorial de un número entero positivo que se introduce por teclado.
8. Escribe un programa que calcule x^n , siendo x y n dos números enteros que se introducen por teclado.
9. El máximo común divisor de los enteros x y y es el entero más grande que es divisor exacto de x y de y . Escribe una función recursiva gcd que devuelva el máximo común divisor de x y y . El máximo común divisor de x y y se define recursivamente como sigue: si y es igual a 0, entonces gcd(x , y) es x ; en caso contrario, gcd(x , y) es gcd(y , $x\%y$).

10. Escribe un programa que utilizando una función recursiva calcule la serie de Fibonacci. La serie de Fibonacci se define mediante:

$$a_0 = a_1 = 1 \qquad a_n = a_{n-1} + a_{n-2}$$

Ejemplo: 1,1,2,3,5,8,13,21,....

11. Escribe un programa que calcule el primer elemento de la serie de Fibonacci que sea mayor o igual que un valor introducido por teclado. Utiliza la función recursiva hecha en el ejercicio anterior.
12. Realizar un programa para resolver el juego de las Torres de Hanoi. El juego consiste en tres varillas verticales. En una de las varillas se apila un número indeterminado de discos. Los discos se apilan sobre una varilla en tamaño decreciente. No hay dos discos iguales, y todos ellos están apilados de mayor a menor radio en una de las varillas, quedando las otras dos varillas vacantes. El juego consiste en pasar todos los discos de la varilla ocupada a una de las otras varillas vacantes. Para realizar este objetivo, es necesario seguir estas simples reglas:
- Sólo se puede mover un disco cada vez.
 - Un disco de mayor tamaño no puede descansar sobre uno más pequeño que él mismo.
 - Sólo puedes desplazar el disco que se encuentre arriba en cada varilla.

El movimiento de n discos se puede visualizar en términos de mover sólo $n-1$ discos (y de ahí la recursión) como sigue:

- Pasar $n-1$ discos de la varilla 1 a la 2, usando la varilla 3 como área de retención temporal.
- Pasar el último disco (el más grande) de la varilla 1 a la 3.
- Pasar los $n-1$ discos de la varilla 2 a la 3, empleando la varilla 1 como área de retención temporal.

El proceso termina cuando la última tarea implica pasar $n=1$ disco, esto es, el caso base. Esto se logra transfiriendo el disco sin necesidad de un área de retención temporal.

Escribe una función recursiva con cuatro parámetros:

- El número de discos por transferir.
- La varilla en la que están colocados inicialmente esos discos.
- La varilla a la que debe pasarse esa pila de discos.
- La varilla que se usará como área de retención temporal.

El programa deberá imprimir las instrucciones precisas requeridas para pasar los discos de la varilla inicial a la varilla de destino. Por ejemplo, para pasar una pila de tres discos de la varilla 1 a la varilla 3, el programa deberá imprimir la siguiente serie de movimientos:

1 → 3
1 → 2
3 → 2
1 → 3
2 → 1
2 → 3
1 → 3