

HOJA 3. RECURSIVIDAD

1. Sea la función, $Q(a,b) : \mathbb{N} \times \mathbb{N}^+ \rightarrow \mathbb{N}$, definida como:

$$Q(a,b) = \begin{cases} 0 & \text{si } a < b \\ Q(a-b, b) + 1 & \text{si } b \leq a \end{cases}$$

Programarla recursivamente. ¿Qué calcula esta función?

2. Escribir una versión recursiva del algoritmo de Euclides, para encontrar el máximo común divisor de dos números enteros.

$$mcd(a,b) = \begin{cases} a & \text{si } b = 0 \\ mcd(b, resto(a/b)) & \text{si } b \neq 0 \end{cases}$$

3. Escribe una función recursiva que calcule a^n para cualquier a número real y para cualquier n entero.
4. a. Escribir una **función** recursiva que dado un número entero positivo calcule el entero obtenido al invertir el orden de sus cifras.
- b. Escribir una **acción** recursiva que dado un número entero positivo muestre por pantalla el número con las cifras invertidas (cada cifra en una línea distinta).
- c. Escribir una **acción** recursiva que dado un número entero positivo muestre por pantalla el mismo número (cada cifra en una línea distinta).

Para el número 543

Aptdo b	Aptdo c
3	5
4	4
5	3

5. Determinar si una cadena de caracteres (sin espacios en blanco) es o no un palíndromo (su lectura de derecha a izquierda coincide con su lectura de izquierda a derecha).
6. Escribir una acción recursiva que muestre la representación binaria de un número en base decimal.
7. Sea v un vector de tamaño n ,
- Construir una acción recursiva que escriba los elementos de v en orden inverso.
 - Construir una acción recursiva que escriba los elementos de v en el mismo orden.
8. Sea v un vector de tamaño n . Construir subalgoritmos recursivos para calcular:
- El máximo de sus elementos.
 - La suma de sus elementos.
 - El mínimo de sus elementos.
 - La media de sus elementos.

- e) Averiguar si las componentes comprendidas entre las componentes i y j están ordenadas de forma creciente.
 - f) Decidir si alguna de las componentes es suma de las que tiene a su izquierda.
 - g) Averiguar si sus componentes están en progresión geométrica.
9. En todos los apartados deberá explicarse el diseño recursivo utilizado, detallando los siguientes pasos:
- Caso (o casos) triviales de la recursividad.
 - Caso general.
 - Solución algorítmica.
- a) Escribir una función recursiva que calcule la suma de los dígitos de un número entero positivo. (Ejemplo: dado 3475, la función debería calcular $3+4+7+5=19$)
 - b) Se llama *normalización* de un número entero positivo al número entre 1 y 9 obtenido del inicial sumando de manera reiterada los dígitos de éste. Por ejemplo, dado 3475, obtenemos $3+4+7+5=19$, de aquí pasamos a $1+9=10$, y de aquí a $1+0=1$, con lo que la normalización de 3475 es 1. Escribir una función recursiva que calcule la normalización de un número entero positivo (Nota: puede usarse la función del apartado a))
 - c) Se llama *índice de normalización* de un número entero positivo al número de llamadas que es necesario realizar a la función del apartado anterior para obtener la normalización. Se entiende que para números de un sólo dígito es necesaria una llamada, y por tanto su índice de normalización es 1. En el ejemplo del enunciado anterior, el índice de normalización del 3475 es 4. Escribir una función recursiva que calcule el índice de normalización de un número entero positivo (Nota: puede utilizarse la función del apartado a))
 - d) Escribir una acción recursiva que calcule a la vez la normalización y el índice de normalización, sin utilizar las funciones de los apartados b) y c).
10. Dados dos números reales a y b (con $a \geq 1$), determinar si el segundo es potencia del primero.
11. Sea v vector ordenado de enteros, n su dimensión ($1 \leq n \leq \text{MAX}$) y x un número entero. Construir los siguientes subalgoritmos:
- a) Búsqueda binaria recursiva, determinando la posición del número entero en el vector. Si el número no está, el subalgoritmo asigna el valor FALSO a uno de sus parámetros.
 - b) Búsqueda secuencial recursiva, determinando la posición del número entero en el vector. Si el número no está, el subalgoritmo asigna el valor FALSO a uno de sus parámetros.
12. Escribir en C++ subprogramas recursivos para:
- a) Contar las vocales de una cadena de caracteres.
 - b) Averiguar si un texto es palíndromo.