

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
Lista de Ejercicios: Recursión
(2016-1)

Horario 0581: prof. Fernando Alva

-
1. Implementar una función recursiva en C que calcule la multiplicación de dos números enteros. Recordar considerar todos los casos: números positivos, negativos y ceros.
 2. Implementar una función recursiva en C que calcule la potencia de un número (**a** elevando a la **b**).
 3. Implementar una función recursiva en C para encontrar el MCD de dos números usando el algoritmo de Euclides:
 - $\text{MCD}(m, n) = n$ si $n \leq m$ y n divide a m
 - $\text{MCD}(m, n) = \text{MCD}(n, m)$ si $m < n$
 - $\text{MCD}(m, n) = \text{MCD}(n, \text{residuo}(m/n))$
 4. Implementar funciones recursivas en C para que, dado un número N , se escriban los números enteros de forma ascendente hasta N (de 1 a N), o descendente desde N (de N a 1).
 5. Implementar una función recursiva en C que permita imprimir las cifras de un número entero de forma invertida.
 6. Implementar una función recursiva en C que enuncie la función de Ackermann, cuya definición es la siguiente:
 - $A(0, n) = n + 1$
 - $A(m, 0) = A(m - 1, 1)$ para $m > 0$
 - $A(m, n) = A(m - 1, A(m, n - 1))$ para $m, n > 0$
 7. Implementar una función recursiva en C que permita determinar la cantidad de veces que una cadena se encuentra contenida en otra. Por ejemplo: sean $\text{cad1} = \text{"mar"}$; $\text{cad2} = \text{"martín armará el camarote el martes"}$; la función debe devolver 4.
 8. Implementar una función recursiva en C que permita determinar si un arreglo de números enteros es palíndromo o no. No podrá usar arreglos adicionales y la función solo podrá recibir dos parámetros: el arreglo y su longitud.
 9. Implementar una función recursiva en C que calcule la moda de un arreglo de números enteros.
 10. Utilizando únicamente funciones recursivas, implemente en C las funciones de manejo de cadenas:
 - `Inserta(cadena1, cadena2, pos)`, donde `cadena2` es la cadena que se va a insertar en `cadena1` desde la posición `pos`.
 - `EliminaAparicion(cadena1, cadena2)` que elimina **todas** las apariciones de `cadena2` en `cadena1`.
 - `MinusculasComunes(cadena1, cadena2)` que devuelve la cadena formada por las letras minúsculas comunes a las cadenas pasadas como parámetros.

NOTA: Toda función adicional también deberá ser recursiva. No debe usar estructuras iterativas en ninguna de las funciones.

11. Implementar una función recursiva que teniendo como únicos parámetros un arreglo bidimensional de enteros (el cual representa una matriz cuadrada de orden “N”) y el orden de la misma, permita obtener el mismo arreglo pero que contenga la transpuesta de dicha matriz. Nota: Se deberá trabajar únicamente con el arreglo bidimensional de enteros, sin utilizar ninguna estructura de datos adicional.
12. Se dispone de una matriz bidimensional de $m \times n$ de tipo carácter, en cuyas casillas solo hay dos posibles valores: ‘X’ o ‘ ’ (espacio en blanco), como se muestra en la Figura:

	1	2	3	4	5	6	7	8	9
1									
2	X			X				X	
3		X						X	
4		X				X			
5							X		
6		X	X				X		X
7								X	X

Las ‘X’ representan entradas, las cuales se pueden conectar en forma horizontal, vertical o diagonal formando lo que se denomina un **cluster**. Dada una coordenada se podría determinar el tamaño del cluster. Por ejemplo, en la figura, dada la coordenada (2, 1) el tamaño del cluster es 3, ya que la zona del cluster lo constituyen los datos de las coordenadas (2, 1), (3, 2) y (4, 2). Para la coordenada (5, 7) el tamaño del cluster sería 6. En conclusión, dada una coordenada, el tamaño del cluster viene determinado por la posición de la misma coordenada más las entradas de sus vecinos, teniendo en cuenta que cada entrada es contabilizada sólo una vez.

Implemente un programa en C que lea una matriz como la de la figura y una coordenada, e invoque a una función recursiva que, con esta información, calcule el tamaño del cluster.