

ALGORITMOS TUPAR

Práctico No 1. Recursión

Implemente las siguientes funciones de manera recursiva e iterativa

1. Programe una función que calcule la n-ésima potencia de un número x.
2. Programe una función que calcule el n-ésimo elemento de la serie de Fibonacci.
3. A partir de un número en decimal, detectar su representación en binario (dividir por 2 hasta que el resultado sea 0 o 1 e ir quedándose con el resto).
4. Implementar una función que calcule el Máximo Común Divisor de dos números.

Implemente de manera recursiva

1. Calcular el número combinatorio de m tomados de a n.
2. Dada una cadena de caracteres, devolver si es o no capicúa.

Arreglos

1. Programe una función recursiva que devuelva la cantidad de veces que un elemento ocurre en un arreglo unidimensional.
2. Programe una función recursiva que ordene los elementos del arreglo de menor a mayor.
3. Implemente un método recursivo que invierta los números de un arreglo de enteros.

Árboles binarios:

1. Realice un programa recursivo que calcule la frontera de un árbol binario, donde por frontera se entiende la lista formada por los elementos almacenados en las hojas del árbol tomados de izquierda a derecha (almacenarlos en un arreglo o imprimirlos por pantalla).
2. Realice programas recursivos para calcular las siguientes propiedades de los árboles:
 - a. Cantidad de nodos total del árbol.
 - b. Cantidad de hojas.
 - c. Altura de un árbol: cantidad de nodos de la rama más larga.
3. Dados dos árboles binarios, programe una función que determine si son isomorfos (misma forma aunque distinto contenido).

Ejercicios adicionales

Implemente de manera iterativa

1. Escriba una función en la que se introduzcan 10 enteros y determine cuáles de estos enteros son pares y cuáles son impares.
2. Escriba una función que tome un valor entero y regrese el número con los dígitos invertidos. Por ejemplo, dado el número 7631, la función deberá regresar 1367.
3. Desarrolle un programa que, para una cierta cantidad de dinero (en pesos), da el cambio apropiado, en billetes o monedas (dar siempre la menor cantidad de billetes/monedas).
4. Convierta una cantidad de tiempo (en segundos, considerar un número entero de segundos), en la correspondiente en horas, minutos y segundos, con arreglo al siguiente formato: 3817 segundos = 1 horas, 3 minutos y 37 segundos
5. Pedir un número natural n y dar sus divisores.
6. Número perfecto es el que es igual a la suma de sus divisores, excluido el mismo.
Ejemplo: $6 = 1 + 2 + 3$ Contraejemplo: $12 = 1 + 2 + 3 + 4 + 6$.
 - a. Escriba un programa que dé los números perfectos de 1 a 200.
 - b. Desarrolle un programa que busque el primer número perfecto a partir de un cierto entero dado por el usuario haciendo uso de la función lógica `EsPerfecto`, que a su vez se apoya en la función `SumCifras` definida en el ejercicio anterior.