

Guía de Ejercicios Prácticos

A continuación se plantean una serie de problemas. Para resolverlos se deberá desarrollar una función recursiva.

1. Implementar una función que permita obtener el valor en la sucesión de fibonacci para un número dado.
2. Implementar una función que calcule la suma de todos los número enteros comprendidos entre cero y un número entero positivo dado.
3. Implementar una función para calcular el producto de dos números enteros dados.
4. Implementar una función para calcular la potencia de dado dos números enteros, el primero representa la base y segundo el exponente.
5. Dada una secuencia de caracteres, obtener dicha secuencia invertida.
6. Desarrollar un algoritmo que permita calcular la siguiente serie $h(n) = 1 + 1/2 + 1/3 + \dots + 1/n$.
7. Desarrollar un algoritmo que permita convertir un número entero en sistema decimal a binario.
8. Implementar una función para calcular el logaritmo entero de número n en una base b , recuerde que $\log_b(n/b) = \log_b n + \log_b b$.
9. Desarrollar un algoritmo que cuente la cantidad de dígitos de un número entero.
10. Desarrollar un algoritmo que invierta un número entero sin convertirlo a cadena.
11. Desarrollar el algoritmo de euclides para calcular el mcd (máximo común divisor) de un número entero.
12. Desarrollar el algoritmo de euclides para calcular el mcm (mínimo común múltiplo) de un número entero.
13. Desarrollar un algoritmo que permita realizar la suma de los dígitos de un número entero, no se puede convertir el número a cadena.
14. Desarrollar una función que permita calcular la raíz cuadrada entera de un número entero, puede utilizar una función auxiliar para que la función principal solo reciba como parámetro en número a calcular su raíz.
15. Implementar un función recursiva que permita recorrer una matriz y mostrar sus valores.
16. Dada una lista de valores desarrollar el algoritmo *quicksort*, que permita ordenar dicha lista.
17. Dada una lista de valores ordenadas previamente a partir del algoritmo del punto anterior, desarrollar un algoritmo que modifique el método de búsqueda binaria para que funcione de forma recursiva, y permita determinar si un valor dado está o no en dicha lista.
18. Salida del laberinto. Encontrar un camino que nos permita salir de un laberinto definido en una matriz de $N \times N$, solo podemos movernos de a una casilla a la vez y que la misma sea adyacente y no esté marcada como pared. Se comenzará en la casilla (0, 0) y se termina en la (N-1, N-1). Nos movemos a la siguiente casilla si es posible, cuando no podemos avanzar retrocedemos sobre nuestros pasos en busca de un camino alternativo.
19. En el momento de la creación del mundo, los sacerdotes del templo de Brahma recibieron una plataforma de bronce sobre la cual había tres agujas de diamante. En la primera aguja estaban apilados setenta y cuatro discos de oro, cada una ligeramente menor que la que estaba debajo. A los sacerdotes se les encomendó la tarea de pasarlos todos desde la primera aguja a la tercera, con dos condiciones, sólo puede moverse un disco a la vez, y ningún disco podrá ponerse encima de otro más pequeño. Se dijo a los sacerdotes que,

cuando hubieran terminado de mover los discos, llegaría el fin del mundo. Resolver este problema de la Torre de Hanoi.

20. Dada la siguiente fórmula de sucesión recursiva, $f(n)$: si $(n=1)$ entonces $f(n)= 3$, y si $(n>2)$ entonces $f(n) = (n - 1) + 2$, realizar una función recursiva que permita calcular el valor de un determinado número en dicha sucesión.
21. Desarrollar un función recursiva que permita calcular el método de la bisección de una función $f(x)$.
22. Desarrollar un función recursiva que permita calcular el método de la secante de una función $f(x)$.
23. Desarrollar un función recursiva que permita calcular el método de Newton-Raphson de una función $f(x)$.