



Prof. Dr. Ivairton Monteiro Santos

Programação de Computadores

Lista de Recursividade

Algoritmo 1

Escreva um algoritmo que leia do usuário um valor inteiro positivo n e então que utilize uma função recursiva que conte/imprima do valor lido n até 1 (decrescente).

Algoritmo 2

Escreva um algoritmo que leia do usuário um valor inteiro positivo n e então que utilize uma função recursiva que conte/imprima de 1 até o valor lido n (crescente).

Algoritmo 3

Escreva um algoritmo que leia dois valores inteiros positivos do usuário e que utilize uma função recursiva para somar os valores (lembre-se que a soma entre dois valores consiste da quantidade de unidades do primeiro valor mais a soma sucessiva de cada unidade do segundo valor – como se fosse um contar de pedrinhas).

Algoritmo 4

Escreva um algoritmo que leia dois valores inteiros positivos do usuário e que utilize uma função recursiva que multiplique os valores (lembre-se que a multiplicação entre dois valores consiste da soma sucessiva do primeiro valor em quantidade equivalente ao segundo valor).

Algoritmo 5

Escreva um algoritmo que leia do usuário um valor inteiro positivo e que calcule o fatorial deste número por meio de uma função recursiva.

Algoritmo 6

A sequência de Fibonacci é definida pela função

$$f(n) = f(n - 1) + f(n - 2)$$

onde

$$f(0) = 0$$

$$f(1) = 1$$

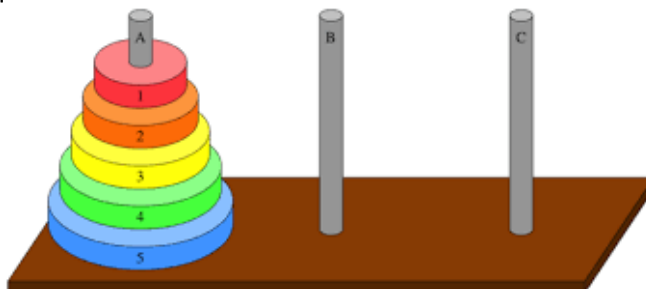
Escreva um algoritmo que leia do usuário um valor inteiro maior, ou igual, a zero e que apresente o valor correspondente da sequência de Fibonacci, conforme formulação apresentada a cima, calculado por meio de uma função recursiva.

Algoritmo 7

Considere um vetor de números inteiros com capacidade de 10 valores. Escreva um algoritmo que leia os valores do usuário e armazene no vetor e então implemente um procedimento recursivo que imprime os valores do vetor na ordem inversa (da última posição para a primeira).

Desafio para os “fortes”! 😊

A Torre de Hanoi é um jogo onde discos de diferentes tamanhos, ordenados do menor para o maior (o maior fica na base) devem ser transferidos um-a-um de uma torre A para uma torre B, com a possibilidade de usar uma torre auxiliar C. No jogo, não é permitido que um disco maior (maior valor) seja posicionado sobre um disco menor, apenas qualquer um menor sobre um maior.



Esse jogo pode ser resolvido por meio de uma função recursiva. A implementação do algoritmo recebe como entrada um valor n , que corresponde à quantidade de discos e então a função recursiva realiza os movimentos dos discos (valores) de uma torre para a outra, utilizando a torre auxiliar. As torres podem ser representadas por um vetor.

Pense e tente implemente o algoritmo que recebe a quantidade de discos e que utilize a função recursiva que realiza os deslocamentos dos discos.

Dica: A função recursiva irá receber como parâmetros o número do disco em processamento e a referência das três torres. O critério de parada ocorre quando a função estiver processando o último disco (o menor deles, valor 1, para a torre B), enquanto isso deverá executar as ações:

Função_desloca_torre(disco – 1, torreA, torreAux, torreB)

Desloca_disco(torreA, torreB)

Função_desloca_torre(disco – 1, torreAux, torreB, torreA)