

Programação II

Lista de Exercícios para aplicação de conceitos básicos de recursividade.

Obs.: Para avaliar cada função, construa uma pequena aplicação para testes.

Construa funções recursivas para cada uma das funcionalidades pedidas.

1. Calcular a soma dos elementos de uma lista numérica.
2. A quantidade total de permutações sobre um conjunto de n elementos é $n!$ (n fatorial). Por exemplo, para o conjunto formado por $\{a,b,c\}$ teremos as seguintes permutações: $\{(a,b,c), (a,c,b), (b,a,c), (b,c,a), (c,a,b), (c,b,a)\}$. É relativamente fácil observar que podemos descrever o conjunto de permutações como 'a' seguido das permutações de 'b' e 'c'; 'b' seguido das permutações de 'c' e 'a'; c seguido das permutações de 'a' e 'b'. Construa uma função recursiva que retorne o conjunto de todas as permutações de um conjunto de entrada de 3 elementos. Como generalizar a função para um conjunto de tamanho k elementos quaisquer ?
3. Calcular o produto de 2 números, x e y . (pesquise o conceito de produto)
4. Calcular a divisão de 2 números, x e y . (pesquise o conceito de divisão)
5. Calcular a raiz quadrada de um número n com tolerância máxima t . (pesquise a definição de raiz quadrada)
6. Pesquisar a existência do elemento e na lista L . Retorna True caso exista, False caso contrário.
7. Inverter uma string de entrada.
8. Testar se um número n passado como parâmetro é um número natural. (pesquise as propriedades de um número natural)
9. Calcular o maior valor de uma lista de números fornecida como entrada.
10. Calcular o menor valor de uma lista de números fornecida como entrada.
11. Testar se uma string de entrada é um palíndromo. Retorna True caso seja, False caso não seja um palíndromo.
12. É possível construir uma função recursiva para converter um valor em base dez para binário ? Tente construir esta função a partir do algoritmo clássico de conversão decimal binário.

Fim.