

CMP1054 - Estrutura de Dados I

2ª Lista de Exercícios - Recursividade

Max Gontijo de Oliveira

- Todas as funções criadas nas questões deverão ser testadas em um programa principal (`main`).
- Caso haja necessidade, crie parâmetros adicionais para sua função recursiva além dos explicitamente solicitados nas questões.

1. Crie uma função recursiva em C++ que retorne a soma dos elementos de um vetor passado por parâmetro.
2. Crie uma função recursiva em C++ que receba por parâmetro um número inteiro positivo N e escreva na tela esse número na base binária.
3. Crie uma função recursiva em C++ que receba por parâmetro um número inteiro positivo N e retorne o seguinte cálculo:

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N} \quad (1)$$

4. Crie uma função recursiva em C++ que receba por parâmetro uma string (vetor de `char`) e que deve devolver a mesma string invertida. Por exemplo, a string "CARRO" será transformada em "ORRAC". Nota: NÃO é para criar uma nova string. É para alterar diretamente a string passada por parâmetro.
5. Crie uma função recursiva em C++ que receba por parâmetro um número inteiro positivo N e escreva na tela, todas as possibilidades de combinações das N primeiras letras do alfabeto. Por exemplo, se o número passado por parâmetro for 3, a saída deverá ser:

ABC
ACB
BAC
BCA
CAB
CBA

Dica: não efetue os testes com números maiores que 5 pois a quantidade de combinações cresce de forma fatorial.

6. Crie uma função recursiva em C++ que receba por parâmetro um vetor de inteiros e retorne o menor elemento do vetor.
7. O problema do caixeiro viajante pode ser descrito da seguinte maneira: sejam N cidades, cada par de cidades possuem uma distância entre si; o problema do caixeiro viajante consiste em encontrar o menor caminho de modo que o caixeiro viajante passe por todas as cidades uma única vez e retorne para a cidade de origem.

Para resolver esse problema pode-se utilizar uma matriz quadrada $A_{N \times N}$ onde cada célula A_{ij} armazena o valor da distância entre a cidade i e a cidade j . Obviamente, $A_{ij} = A_{ji}$ para todo i ; e a diagonal principal da matriz onde $i = j$ sempre será igual à 0 e não terá nenhuma relevância.

Crie uma função recursiva que receba essa matriz $A_{N \times N}$ com as distâncias já preenchidas e um vetor V_N que deverá ser usado para armazenar o caminho com menor distância total (índices das cidades na ordem em que devem ser visitadas).