

== Lista de Exercícios 4 ==

Em todos os exercícios abaixo, utilize bibliotecas e funções do C++.

1. Escreva um programa para calcular o elemento i da série de Fibonacci. Divida o código do programa em dois arquivos com código fonte (*.cpp), um para a função principal e outro para a função que calcula a série, e um arquivo de interface (*.h).
2. Escreva um programa que calcule a combinação de n elementos tomados m a m . Semelhantemente ao Exercício 1, o código deve ser dividido em dois arquivos com código fonte e um arquivo de interface.
3. Escreva um programa para inserir elementos em uma lista encadeada. Os elementos devem ser inseridos no final da lista. Semelhantemente ao Exercício 1, o código deve ser dividido em dois arquivos com código fonte e um arquivo de interface.
4. Escreva um programa para inserir elementos em uma árvore binária. Estabeleça um critério para inserir os elementos. Semelhantemente ao Exercício 1, o código deve ser dividido em dois arquivos com código fonte e um arquivo de interface.
5. Crie uma classe `Fibonacci` para calcular o elemento i da série de Fibonacci. Essa classe contém dois (2) métodos públicos e um (1) privado e dois (2) atributos privados inteiros (`numero` e `resultado`). Um dos métodos públicos é uma função do tipo "set" que é utilizada para inicializar e atualizar o atributo `numero`. Esse método deve invocar a função privada `calculaFibonacci` que, por sua vez, calcula o elemento i da série e atualiza o valor armazenado no atributo `resultado`. A função do tipo "get" é usada para recuperar o valor encontrado. Escreva todas as funções e ainda o programa principal que utiliza todos os métodos da classe implementada.
6. Repita o Exercício 5, utilizando o construtor da classe `Fibonacci` para inicializar o atributo `numero` ao invés do método "set".
7. Escreva a classe `Arvore` e implemente métodos públicos para inserção e busca em uma árvore binária. Gere uma biblioteca compartilhada chamada `libtree.so`. Implemente um programa que insira elementos em uma árvore binária e, depois disso, realize uma busca por um elemento não existente. O programa deve utilizar a biblioteca criada `libtree.so`.