
1ª Lista de Exercícios – Revisão

Data Entrega: vide site

Enviar e-mail:

- Para: ed2tarde@marciobueno.com (**para a turma da tarde**) ou ed2noite@marciobueno.com (**para a turma da noite**)
- Assunto: Lista1
- Anexo: Um único arquivo .ZIP com todos os arquivos .cpp, .h e arquivos de projeto do MS Visual C++

OBS: e-mails fora deste padrão serão desconsiderados

1. Considere uma árvore binária de busca sem repetição cujos nós armazenarão um tipo definido através de um **typedef**. Construa os procedimentos e funções abaixo. Crie os arquivos: “ArvoreBinaria.h” que contém os **structs** e protótipos das funções e procedimentos necessários, “ArvoreBinaria.cpp” que contém as implementações das funções e procedimentos definidos no arquivo “ArvoreBinaria.h”, e “Main.cpp” que contém um main que cria um menu para o usuário acessar todos os procedimentos e funções implementados.
 - a. Um procedimento para inicializar uma árvore vazia
 - b. Uma função recursiva para inserir um novo nó que retorne um booleano (**bool**), que será **true** se o elemento foi realmente inserido e **false**, caso contrário
 - c. Uma versão não recursiva da função descrita no item **b**
 - d. Uma função recursiva para contar o número de folhas
 - e. Uma versão não recursiva da função descrita no item **d**
 - f. Um procedimento não recursivo para percorrer a árvore utilizando o passeio em-ordem
 - g. Um procedimento não recursivo para percorrer a árvore utilizando o passeio em pré-ordem
 - h. Um procedimento não recursivo para percorrer a árvore utilizando o passeio em pós-ordem
 - i. Uma função recursiva que determina a frequência de um item numa árvore binária de busca
2. Faça uma função que insira de 2 a 15 valores aleatórios não repetidos em uma árvore binária (não de busca), verifique se ela é ou não de busca (utilize os dois algoritmos vistos em sala – utilizando as funções **menor** e **maior** de árvore binária e de árvore binária de busca) e retorne **true** se estas duas versões derem o mesmo resultado e **false**, caso contrário. Crie um **main** que execute esta função 1.000.000 vezes e exiba na tela SUCESSO se todas as execuções for **true** e FALHA, caso contrário.