

Nome: _____ Matrícula: _____

Leia atentamente as instruções abaixo:

- Fazer o download do arquivo `Avaliacao3EDLab2-2020-1.zip` do *site* do curso e descompactá-lo na sua máquina. Este arquivo contém todos os códigos para o desenvolvimento da prova.
- A resposta de cada questão deve, **obrigatoriamente**, estar entre cada par de marcadores (`//Qi`, `//-Qi`). Assim, a questão 1 está entre `//Q1` e `//-Q1`, a questão 2 entre `//Q2` e `//-Q2` e assim por diante. Não remover, em hipótese alguma, tais marcadores de questões da sua prova. Caso sua solução tenha mais de uma função ou operação, elas devem estar entre esses marcadores.
- A prova é **individual**. Respostas iguais terão notas iguais e **iguais a zero**.
- Existe apenas um projeto do Code::Blocks que será usado na prova.
- Ao terminar a avaliação, enviar ao servidor – usando a janela de upload – cada arquivo de código que contém as respostas das questões da sua prova. Aguarde um momento e verá as suas respostas de cada questão da prova.
- **O desenvolvimento e envio do código são de inteira responsabilidade do aluno!**
- Endereço do servidor: `http://edlab2ufjf.uk.to:8080/edlab2ufjf/`

Questões:

1. (30 Pontos) Desenvolver a operação `int* ArvBinBusca::criaVetFolhas(int k, int *n)` que, dado um nível `k`, aloque (com um número de elementos apropriado), preencha e retorne um vetor com os valores de todas as folhas existentes até o nível `k` de uma árvore binária. Se a árvore for vazia, retornar `NULL`. A operação deve armazenar o tamanho do vetor que foi alocado no ponteiro `n`. Percorrer a árvore uma única vez, fazendo um percurso em pré-ordem, e não visitar nós de níveis desnecessários. Completar as posições não utilizadas do vetor com o valor `-1`.
2. (35 Pontos) Considerando uma árvore binária de busca (ABB):
 - (a) Desenvolver a operação `int ArvBinBusca::classificaFilhos(NoArv *pai)` para, dado o ponteiro `pai` para um nó válido, classificar este nó quanto aos seus filhos. Esta operação deve retornar `-1`, `0`, `1` ou `2` caso o nó `pai` tenha, respectivamente, apenas um filho à esquerda, zero filhos (folha), apenas um filho à direita ou dois filhos.
 - (b) Usando a operação `classificaFilhos()`, desenvolver uma nova operação **não recursiva** `int ArvBinBusca::contaNos1Filho(int val)` para contar e retornar quantos nós possuem apenas um filho no caminho que vai da raiz até o nó de valor `val` em uma ABB. O nó que possui o valor `val` (se existir) deve ser levado em consideração.
3. (35 Pontos) Desenvolver a operação auxiliar `NoArv* ArvBinBusca::auxRemoveMenor(NoArv *p, int val)` que percorre **recursivamente** uma ABB, removendo todos os nós cuja informação armazenada seja menor do que a chave `val` indicada no parâmetro. Para auxiliar nesta implementação, utilize a operação `libera()` sempre que possível. Usar, obrigatoriamente, a propriedade fundamental de uma ABB para visitar o número mínimo de nós.