

Nome: _____ Matrícula: _____

Leia atentamente as instruções abaixo:

- Fazer o download do arquivo Avaliacao3EDLab2-2017-3.zip do site do curso. Descompactá-lo em um diretório de 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, obrigatoriamente.
- Colocar no arquivo main.cpp seu nome completo e número de matrícula.
- A prova é **individual e sem qualquer tipo de consulta**.
- Existe apenas um projeto do Code::Blocks que será usado na prova.
- Antes de sair do laboratório, enviar ao servidor – usando a janela de upload – cada arquivo de código que contém as respostas das questões da sua prova.
- **O desenvolvimento e envio do código são de inteira responsabilidade do aluno!**
- Endereço do servidor: <http://172.18.41.97:8080/edlab2ufjf/>

1. (35 Pontos) Considerando o TAD de árvore binária de busca (ABB), desenvolver uma operação, **não recursiva**, para retornar o nível do nó que possui o maior valor entre todos nós que formam o caminho de busca da chave `ch` em uma ABB. Caso a árvore seja vazia retornar `-1`. Observe que, se a árvore não for vazia, sempre existirá tal caminho. Protótipo:

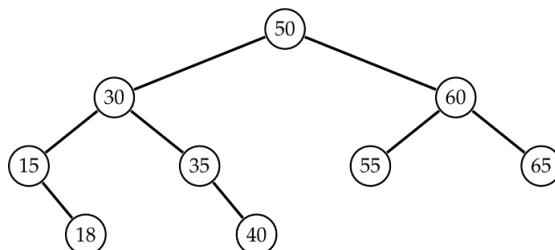
```
int ArvBinBusca::nivelMaiorValor(int ch);
```

2. (35 Pontos) Desenvolver uma operação para calcular e retornar a diferença entre as informações contidas nas folhas com o maior valor e com o menor valor de uma árvore binária (AB). Retornar `-1` se a árvore for vazia. Protótipo:

```
int ArvBinBusca::difMaxMinFolha();
```

3. (30 Pontos) Dados um intervalo fechado $I = [a, b]$ e o TAD de árvore binária de busca, desenvolver uma operação para calcular e retornar a altura h do intervalo I . A altura h de I corresponde à altura da subárvore formada por todos os nós dentro do intervalo I . Como se sabe, a altura de uma árvore vazia é `-1`. Usar a propriedade de ABB para visitar o número mínimo de nós da árvore binária de busca. Protótipo:

```
int ArvBinBusca::alturaIntervalo(int a, int b);
```

Exemplo:*Saída:*

```

Questão 1: m = arv.nivelMaiorValor(55)
Saída: nível: 1 (maior valor 60)
Questão 2: d = arv.difMaxMinFolha()
Saída: diferença: 47 (resultado de 65-18)
Questão 3: h = arv.alturaIntervalo(10,45)
Saída: altura: 2
  
```