

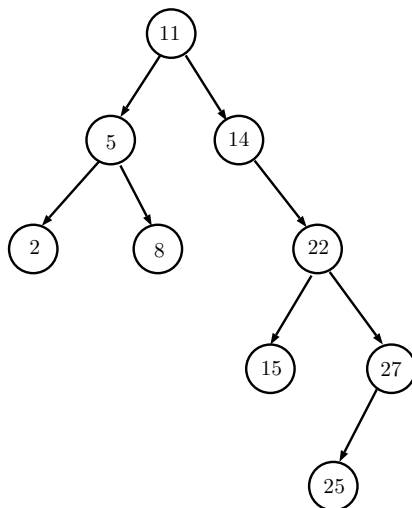
Modalidade: Em dupla ou individual.

Data de entrega: 05/08/2019 até as 23:59 hs no e-mail algestruturados.unioeste@gmail.com . Entregar apenas os códigos fontes **descompactados**, informando os nomes dos alunos. São descontados 25% da nota por dia de atraso.

Fazer um programa em linguagem C para gerenciar o cadastro de uma biblioteca, implementando as estrutura de dados tal que as informações sejam armazenadas em arquivos binários. Cada livro é identificado por um código (inteiro). Também são armazenados o título do livro, autor e quantidade de exemplares no acervo. O cadastro é gerenciado usando uma estrutura de árvore binária tendo o código do livro como chave. O programa deve oferecer ao usuário um menu com as seguintes opções:

- **Carregar arquivo de inicialização:** carrega as informações de um arquivo texto contendo uma lista de livros, conforme sintaxe especificada mais abaixo. As informações devem ser inseridas em arquivo(s) binário(s), zerando-o(s) caso haja informações prévias.
- **Inserir livro:** dados código, título, autor e quantidade de exemplares, inserir o livro no cadastro
- **Atualizar número de exemplares:** permite atualizar o número de exemplares a partir do código do livro
- **Remover livro:** dado o código do livro, remove-o definitivamente do cadastro
- **Buscar dados do livro:** dado o código do livro, retorna suas informações. Caso não exista, imprimir a mensagem “livro não encontrado”.
- **Imprimir Árvore Binária:** imprime a árvore binária apenas com os códigos dos livros de acordo com a notação [*raiz*, *esq*, *dir*]. Assim uma árvore como a da figura abaixo irá gerar a saída:

[11, [5, [2, [], []], [8, [], []]], [14, [], [22, [15, [], []], [27, [25, [], []], []]]].



- **Imprimir Árvore Binária por Nível:** imprime a árvore binária apenas com os códigos dos livros, por nível. Assim uma árvore como a da figura anterior irá gerar a saída:

```
11
5 14
2 8 22
15 27
25
```

- **Imprimir Acervo:** imprime na console todas as informações dos livros cadastrados, ordenados por código em ordem crescente.
- **Gerar Listagem de livros ordenados por título:** gera um arquivo texto com as informações dos livros ordenados lexicograficamente por título. A ordenação deve ser feita em memória usando o algoritmo *Quicksort*, e depois o resultado gravado no arquivo texto, sendo um livro por linha.

Para as operações de inserção, alteração, remoção e busca, apenas o registro que precisa ser manipulado em cada momento é que deve ser carregado na memória principal. Ou seja, **NÃO** carregar todo o conteúdo do(s) arquivo(s).

No arquivo texto de carga, as informações de um livro são identificadas por:

`< código > ; < título > ; < autor > ; < número – de – exemplares >`

sendo um por linha. O caracter ; (ponto-e-vírgula) é usado para separar cada informação.

O programa deve ser capaz de detectar e eliminar espaços em brancos desnecessários contidos no arquivo texto sendo carregado.

Exemplo de arquivo texto para entrada:

```
70;0 Senhor dos Anéis; J. R. R. Tolkien;5
30;As Brumas de Avalon; Marion Zimmer Bradley;2
150;Sapiens: uma breve história da humanidade;Yuval Harari;11
20;Princípios e Práticas de Programação com C++; Bjarne Stroustrup;3
10;Programming in Prolog; W. Clocksin;2
90;Primeiras Estórias; João Guimarães Rosa;4
111;Memorial de Maria Moura; Rachel de Queiroz;5
6;Memórias Póstumas de Brás Cubas; Joaquim Maria Machado de Assis;7
45;0 Discurso do Método; René Descartes;3
200;Ulisses; James Joyce;7
```

Quanto ao uso de arquivos binários para armazenamento das informações:

- devem ser criados 2 arquivos binários distintos: um para conter as informações dos livros e outro para conter a estrutura da árvore binária
- os arquivos deve ter estrutura similar ao mostrado na implementação de lista em arquivo (ver slides!)
- o arquivo que contém a estrutura da árvore binária deve ter um cabeçalho gravado no seu início com as seguintes informações:
 - **raiz:** número do registro que corresponde à raiz da árvore binária
 - **livre:** número do registro que corresponde à cabeça da lista de nós livres (liberados após remoção)
 - **topo:** número do primeiro registro livre no fim do arquivo
- o arquivo das informações dos livros deve ter um cabeçalho gravado no seu início contendo:
 - **livre:** número do registro que corresponde à cabeça da lista de registros livres (liberados após remoção)
 - **topo:** número do primeiro registro livre no fim do arquivo
- ao fazer a remoção de um livro, a posição do registro removido deve ser encadeada na lista de registros livres do arquivo das informações dos livros, e similarmente o registro correspondente ao nó da árvore binária deve ser encadeado na lista de nós livres do arquivo da árvore binária
- ao fazer inserção de um livro deve ser verificado se há registros livres para serem reaproveitados. Caso não haja, deve ser usado o registro indicado pela variável **topo** no cabeçalho do arquivo correspondente.

O trabalho deverá ser apresentado oralmente para o professor com a presença dos integrantes do grupo, em data a ser marcada. A avaliação do trabalho está condicionada à apresentação oral.

Na avaliação será feita média ponderada, considerando os seguintes pesos:

- Carregar arquivo de inicialização: 3
- Inserir livro: 1
- Atualizar número de exemplares: 1
- Remover livro: 3
- Buscar dados do livro: 1
- Imprimir Árvore Binária: 2
- Imprimir Árvore Binária por Nível: 2
- Imprimir Acervo: 1
- Gerar Listagem de livros ordenados por título: 2
- organização do código (identação, legibilidade e comentários): 1
- emprego adequado das estruturas de dados: peso 2
- apresentação oral: peso 1

Trabalhos copiados ou plagiados receberão nota ZERO!