

Modalidade: em grupo (no máximo 2 alunos por grupo)

Data de entrega: 05/Fev/2025 até 23:59 na tarefa do Microsoft Teams. Entregar UM ÚNICO ARQUIVO COMPACTADO .ZIP CONTENDO APENAS OS ARQUIVOS FONTES, e um arquivo *readme* no formato TXT, informando os nomes dos alunos do grupo e instruções para uso do programa. São descontados 25% da nota por dia de atraso.

Fazer um programa em linguagem C para gerenciar o cadastro de livros de uma livraria, utilizando árvore 2-3 implementada em **arquivo binário**, conforme descrito a seguir:

- são armazenadas as seguintes informações de cada livro:
 - **codigo:** número inteiro que identifica univocamente cada livro
 - **titulo:** título do livro, tendo no máximo 150 caracteres
 - **autor:** nome(s) do(s) autor(es), tendo no total, no máximo, 200 caracteres
 - **editora:** nome da editora, tendo no máximo 50 caracteres
 - **edicao:** número inteiro indicando a edição
 - **ano:** ano da publicação/edição
 - **preco:** preço unitário, sendo número real com 2 casas decimais
 - **estoque:** número de exemplares em estoque
- O sistema deve ter as seguintes funcionalidades implementadas, organizadas convenientemente em menus:
 - cadastrar: dados o código, título, autor, editora, número da edição e o ano, insere as informações no cadastro.
 - remover: dado o código, remove as informações do livro correspondente
 - imprimir dados do livro: dado um código de livro, imprimir as informações do respectivo livro
 - listar todos os livros: fazer o percorrimento *in ordem* imprimindo apenas o código e o título do livro
 - imprimir árvore por níveis: imprime as chaves por níveis a partir da raiz

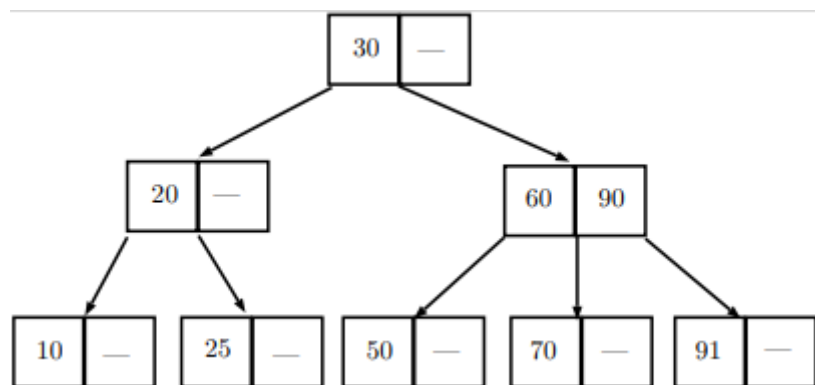


Figura 1: Exemplo de árvore 2-3

Considerando a árvore da Figura 1, a saída seria:

```

[30, -]
[20, -] [60, 90]
[10, -] [25, -] [50, -] [70, -] [91, -]
  
```

- imprimir lista de livros do arquivo de índices: imprime a lista de nós livres da árvore devido às remoções
- imprimir lista de livros do arquivo de dados: imprime a lista de registros livres devido às remoções no arquivo de dados
- calcular total: imprime o número total de livros cadastrados
- realizar operações em lote: realiza operações de inserção, alteração e remoção a partir de arquivo texto a ser carregado, cuja sintaxe é descrito mais adiante.

Toda a interface do programa deve ser apresentada em modo texto, sendo executado no *prompt* do *shell* do sistema operacional.

O arquivo texto a ser carregado, deve ser conforme a sintaxe abaixo, sendo cada campo separado por ; (ponto-e-vírgula):

`< codigo >; < titulo >; < autor >; < editora >; < edicao >; < ano >; < preco >; < estoque >`

Exemplo de conteúdo de arquivo .txt contendo lote de registros a serem incluídos:

```
7;Memorias Postumas de Bras Cubas;Machado de Assis;Bookman;4;2022;25,90;5
11;A insustentavel leveza do ser;Milan Kundera;Abril;3;2015;30,05;7
27;A Hora da Estrela;Clarice Lispector;Abril;5;2007;40,70;3
5;Hamlet;William Shakespeare;Pensamento;20;1998;80,50;7
13;Dom Casmurro;Machado de Assis;Abril;7;1990;20,99;8
8;A condicao Humana;Hannah Arendt;Pensamento;5;2004;50,00;9
20;Sagarana;Guimaraes Rosa;Abril;2;2014;70,99;20
4;As origens do Totalitarismo;Hannah Arendt;Pensamento;3;2018;44,50;7
33;0 Alienista;Machado de Assis;Bookman;7;1996;27,30;28
```

Assim, a linha:

```
7;Memorias Postumas de Bras Cubas;Machado de Assis;Bookman;4;2022;25,90;5
```

é interpretada como:

- o código do livro é 7
- o título do livro é Memoria Postumas de Bras Cubas
- o autor é Machado de Assis
- a editora é Bookman
- a edição é 4
- o ano de publicação/edição é 2022
- o preço unitário é R\$25,90
- há 5 exemplares no estoque

O sistema deve ser capaz de eliminar espaços em branco sobressalentes no início e no fim de cada campo. Por exemplo:

```
7;      Memorias Postumas de Bras Cubas;Machado de Assis      ; Bookman ;4 ;2022; 25,90;5
```

e

```
7;Memorias Postumas de Bras Cubas;Machado de Assis;Bookman;4;2022;25,90;5
```

representam o mesmo livro.

As informações devem ser armazenadas em 2 arquivos binários conforme mostrado na Figura 2:

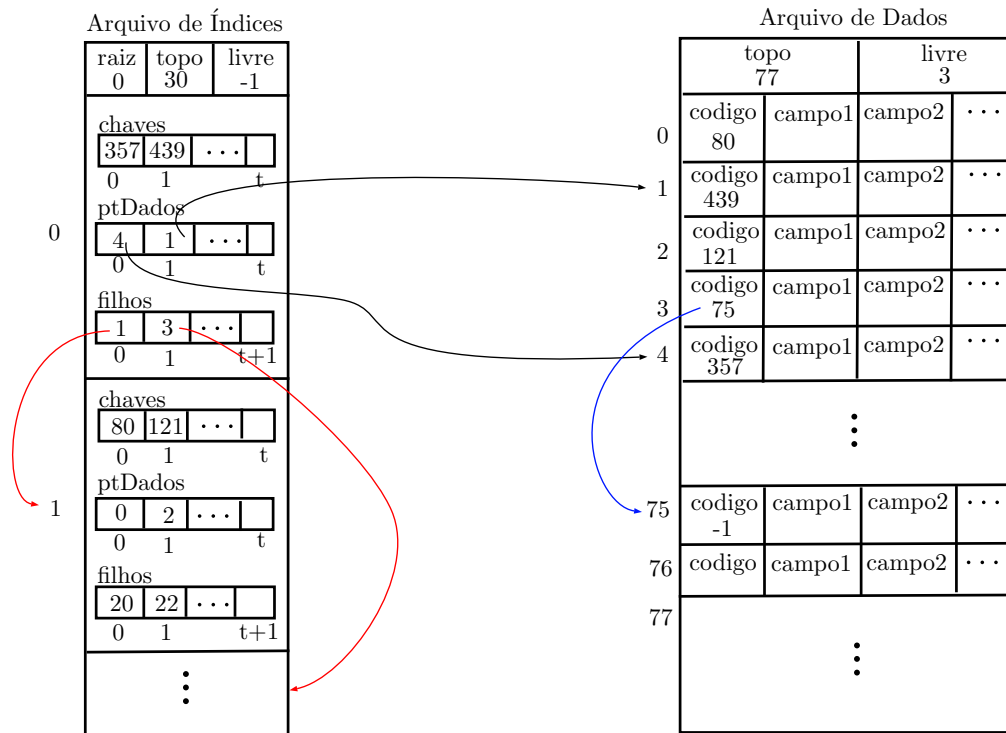


Figura 2: Esquema de arquivos para árvore 2-3

- um arquivo para os registros de dados: contendo o código, título, autor, editora, número da edição e o ano. O campo `código` pode ser aproveitado para o encadeamento de registros livres ou utilizar um campo adicional para tal propósito.
- um arquivo para registros de índices, organizado na forma de árvore 2-3. Além de campos para encadeamento dos nós da árvore, devem haver campos para armazenar as posições dos registros de dados no arquivo de dados. No caso, para árvore 2-3, o esquema apresentado na Figura 2 deve ser: **chaves** são apenas 2, **filhos** são 3, e **ptDados** são 2.

Além disso:

- tanto o arquivo de dados quanto o arquivo de índices devem ser arquivos binários.
- a árvore **NÃO DEVE SER CARREGADA INTEIRAMENTE** na memória principal, apenas as informações do registro sendo manipulado no momento.
- no início do arquivo de índices (árvore 2-3) deve haver um cabeçalho contendo o endereço do registro raiz, a posição do topo (primeira posição livre do arquivo) e o endereço da cabeça de nós (páginas) livres.
- no início do arquivo de dados deve haver um cabeçalho contendo o endereço do topo (primeira posição livre do arquivo) e o endereço da cabeça de registros de dados livres
- no caso de remoção de um livro do cadastro, a posição correspondente dentro do arquivo binário deve ser colocada em uma lista de nós livres para reaproveitamento em futuras inserções. Deve existir uma lista de livres para o arquivo de índices e outra para o arquivo de dados. Pode ser reaproveitado algum campo do registro para fazer o encadeamento da lista de livres.
- o código-fonte (arquivos `.c` e `.h`) deve ser documentado, incluindo para cada função implementada: o propósito da função, pré-condições e pós-condições, conforme mostrado em sala de aula.

Critérios de avaliação:

- o trabalho deverá ser apresentado oralmente em grupo, seguido de arguição sobre o código/teoria relacionada.

- a nota referente à apresentação oral é individual conforme desempenho do aluno na arguição.
 - o aluno do grupo que não esteja presente durante a apresentação oral, não terá a nota correspondente.
- documentação/organização do código-fonte, e readme: peso 1
 - apresentação oral: peso 1
 - implementação das funcionalidades: peso 8

Em todas as funcionalidades, a avaliação será feita considerando corretude e qualidade do código. Trabalhos copiados ou plagiados receberão nota ZERO!