

Algoritmos e Estruturas de Dados 2

Trabalho 1

Prof. Ricardo Oliveira - 2025/1

Enunciado

A Universidade Fictícia de Televisão, Propaganda e Rádio (UFTPR) acabou de aprovar um novo regulamento que trata sobre a aprovação dos alunos em suas disciplinas. Seguindo o novo regulamento:

- Se a nota for maior ou igual a **8,0** e a frequência for maior ou igual a **50%**, aluno **aprovado**; se não:
- Se a nota for maior ou igual a **6,0** e a frequência for maior ou igual a **75%**, aluno **aprovado**; se não:
- Se a nota for maior ou igual a **6,0** e frequência for menor que **75%**, aluno **reprovado por frequência**; se não:
- Se a nota for menor que **6,0** e frequência for maior ou igual a **75%**, aluno **reprovado por nota**; se não:
- Aluno **reprovado por nota e frequência**.

Seu professor está confuso com as novas regras, e pediu sua ajuda. O professor tem um arquivo chamado `diario.txt`. A primeira linha do arquivo tem o número de aulas dadas no semestre. Cada uma das próximas linhas descreve um aluno da turma. Cada linha contém, nessa ordem: Registro Acadêmico (RA); nome; uma *string* binária F onde $F[i]$ é 1 se o aluno esteve presente na aula i , ou 0 se o aluno faltou na aula i ; e a nota do aluno, com exatamente uma casa decimal, separada por vírgula.

Como exemplo, considere o seguinte arquivo `diario.txt`:

```
10
14562 joao da silva 1110111101 7,3
22018 maria joaquina 1100110001 6,4
5135 enzo valentino augusto 1101011100 3,1
2115 leticia magalhaes 1111011011 9,9
14563 joao sauro 1111111101 5,9
678953 turista 0100000010 0,1
28026 pedro 0111011001 8,4
20652 joao paulo 1010100100 10,0
531569 alberto aistein 1111111111 10,0
```

Neste exemplo, foram dadas 10 aulas e a turma tem 9 alunos. O primeiro aluno de exemplo é *joao da silva*, cujo RA é 14562; cuja nota é 7,3; e que veio nas aulas 1, 2 e 3, faltou na aula 4, veio nas aulas 5, 6, 7 e 8, faltou na aula 9 e veio na aula 10. Outros oito alunos são dados nesse exemplo.

Há outros arquivos de exemplos em anexo ao trabalho no *Moodle*.

Escreva um programa em C que lê o arquivo `diario.txt` e carrega suas informações em uma estrutura de dados. Em seguida, seu programa deve ler e processar uma sequência de comandos do usuário. Cada comando pode ser:

- `c [RA]`: consulta o aluno com o RA dado, e imprime, *no formato dado no exemplo abaixo*, uma linha contendo: seu RA; seu nome; sua nota; sua frequência (em porcentagem) e sua situação (dentre `AP`, `REP-F`, `REP-N` e `REP-N-F` para aprovado, reprovado por frequência, reprovado por nota e reprovado por nota e frequência, respectivamente). Caso o RA dado não esteja no diário, imprima `RA [RA] nao encontrado`;

- **i** [A] [B]: imprime, *em ordem crescente do RA e no formato dado no exemplo abaixo*, as informações referentes a todos os alunos cujos RAs estão entre *A* e *B*, inclusive. Após, imprima [T] **alunos encontrados.**, sendo *T* o número de alunos listados neste comando.
- **p** [A]: imprime, em qualquer ordem (mas *no formato dado no exemplo abaixo*), a lista de alunos que estiveram presentes na aula *A*. Após a lista, imprima [T] **presentes na aula [A].**, sendo *T* o número total de alunos presentes nessa aula.
- **e**: imprime, *no formato dado no exemplo abaixo*, as estatísticas da turma (número de alunos aprovados, reprovados por nota, reprovados por frequência e reprovados por nota e frequência).
- **F**: encerra a execução.

Considere o seguinte exemplo de execução (o sinal > indica entrada do usuário), assumindo que o arquivo `diario.txt` é o dado como exemplo acima:

```
> c 14562
(14562) joao da silva 7,3 80.00% [AP]

> c 678953
(678953) turista 0,1 20.00% [REP-N-F]

> c 223344
RA 223344 nao encontrado.

> i 20000 30000
(20652) joao paulo 10,0 40.00% [REP-F]
(22018) maria joaquina 6,4 50.00% [REP-F]
(28026) pedro 8,4 60.00% [AP]
3 alunos encontrados.

> i 20 25
0 alunos encontrados.

> p 1
(14562) joao da silva 7,3 80.00% [AP]
(22018) maria joaquina 6,4 50.00% [REP-F]
(5135) enzo valentino augusto 3,1 60.00% [REP-N-F]
(2115) leticia magalhaes 9,9 80.00% [AP]
(14563) joao sauro 5,9 90.00% [REP-N]
(20652) joao paulo 10,0 40.00% [REP-F]
(531569) alberto aistein 10,0 100.00% [AP]
7 presentes na aula 1.

> p 8
(14562) joao da silva 7,3 80.00% [AP]
(5135) enzo valentino augusto 3,1 60.00% [REP-N-F]
(14563) joao sauro 5,9 90.00% [REP-N]
(20652) joao paulo 10,0 40.00% [REP-F]
(531569) alberto aistein 10,0 100.00% [AP]
5 presentes na aula 8.

> e
4 aprovados;
1 reprovados por nota;
2 reprovados por frequencia;
2 reprovados por nota e frequencia.
```

```

> c 531569
(531569) alberto aistein 10,0 100.00% [AP]

> i 0 1000000000
(2115) leticia magalhaes 9,9 80.00% [AP]
(5135) enzo valentino augusto 3,1 60.00% [REP-N-F]
(14562) joao da silva 7,3 80.00% [AP]
(14563) joao sauro 5,9 90.00% [REP-N]
(20652) joao paulo 10,0 40.00% [REP-F]
(22018) maria joaquina 6,4 50.00% [REP-F]
(28026) pedro 8,4 60.00% [AP]
(531569) alberto aistein 10,0 100.00% [AP]
(678953) turista 0,1 20.00% [REP-N-F]
9 alunos encontrados.

> F

```

Implementação

- Seu trabalho deve **obrigatoriamente** implementar *pelo menos* uma das seguintes estruturas:

- **Árvore AVL**
- **Árvore Rubro-Negra**

Outras estruturas podem ser utilizadas desde que *pelo menos* uma das dadas acima também seja;

- Você deve implementar pelo menos o método de **busca** da estrutura (para responder as consultas) e o método de **inserção de 1 (um) registro** da estrutura (para carregar o arquivo). Os registros devem ser carregados **um por um** chamando esse método;
- Você pode assumir que:
 - O número de aulas é no máximo 60;
 - Não há um limite para o número de alunos;
 - O RA de cada aluno é um inteiro entre 0 e 10^9 inclusive;
 - Não há dois alunos com o mesmo RA;
 - O nome de um aluno é uma *string* com no máximo 30 caracteres. Cada caractere é uma letra minúscula ou um espaço. Há pelo menos uma letra no nome;
 - A nota é um valor entre 0,0 e 10,0 inclusive;
 - O usuário não irá entrar com comandos que não sejam os listados acima.
- Lembre-se sempre de liberar toda a memória alocada ao final da execução do programa.

Orientações

- O trabalho pode ser feito por equipes de *no máximo* 2 (dois) estudantes;
- Submeta, via *Moodle*, um pacote **zip** ou **tar.gz**¹ contendo todo o código-fonte necessário para compilar e executar sua solução, além de um arquivo de texto (txt) onde conste:
 - O nome de todos os integrantes da equipe;
 - Qual(is) estrutura(s) de dados foi(ram) escolhida(s) pela equipe, e por quê;

¹Outros pacotes (rar/zip/etc.) não serão aceitos

- Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- **Não** inclua no pacote arquivos de configuração da sua IDE (arquivos do CodeBlocks/VisualStudio/etc), *sob pena de redução da nota*. Esses arquivos não são importantes para a compilação e execução do código;
- Comente adequadamente seus códigos para facilitar a correção;
- **Atenção:** a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga **fielmente** o formato de entrada e de saída dado nos exemplos*, sob pena de *grande* redução da nota;
- Certifique-se que seu programa compila e funciona antes de submetê-lo;
- O trabalho deve ser entregue até **18 de Maio de 2025, 23:59**, exclusivamente via *Moodle*. Trabalhos entregues por outros meios ou fora do prazo não serão aceitos. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
- Trabalhos detectados como cópia/plágio (de colegas, da internet ou de ferramentas de IA), ou comprados, receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.