

[Prova 01] Estrutura de Dados

Prof.: Jean Nunes

Aluno(a): _____

Instruções:

- Esta avaliação é individual.
- É permitido e recomendado a utilização de códigos desenvolvidos durante as atividades da disciplina.
- O uso de Inteligência Artificial e/ou códigos da *Internet* implicará em um decréscimo de 50% na nota final.
- Organize seu código de forma clara e bem comentada.
- Certifique-se de que seu código compila e executa corretamente.

Materiais Fornecidos:

- **main.c**: Arquivo principal do programa. O programa depende de funções externas (**carregar_alunos**, **ordenar_alunos**, **listar_alunos**, **liberar_memoria**) que devem estar implementadas no arquivo **utils.c**. Ele também utiliza uma estrutura de dados chamada **Aluno**, que deve estar definida no arquivo **utils.h**. O Fluxo Geral de execução do programa é:

1. Recebe o nome do arquivo CSV (`notas.csv`) como argumento.
2. Carrega os dados dos alunos.
3. Ordena e exibe os alunos.
4. Libera os recursos alocados e finaliza.

- **utils.c**: Arquivo com funções utilitárias (**somente as funções vazias devem ser implementadas.**).
- **utils.h**: Arquivo de cabeçalho com declarações de tipos e funções (**esse arquivo não deve ser modificado.**).
- **notas.csv**: Arquivo contendo o banco de dados das notas dos alunos.

Funções a serem implementadas:

```
Aluno *realocar_memoria_aluno(Aluno *alunos, int novo_tamanho);
Avaliacao* realocar_memoria_avaliacao(Avaliacao *avaliacoes, int novo_tamanho);
void adicionar_avaliacoes(Aluno *aluno, Avaliacao avaliacao);
void liberar_memoria(Aluno *alunos, int num_alunos);
void calcular_notas(Aluno *aluno);
```

- **Aluno *realocar_memoria_aluno(Aluno *alunos, int novo_tamanho);**
 - A função **Aluno *realocar_memoria_aluno(Aluno *alunos, int novo_tamanho)** tem como objetivo ajustar dinamicamente o tamanho do array de estruturas do tipo **Aluno**. Ela utiliza alocação dinâmica de memória para expandir o tamanho do array conforme necessário.
- **Avaliacao* realocar_memoria_avaliacao(Avaliacao *avaliacoes, int novo_tamanho);**
 - A função **Avaliacao* realocar_memoria_avaliacao(Avaliacao *avaliacoes, int novo_tamanho)** tem como objetivo ajustar dinamicamente o tamanho do array de estruturas do tipo **Avaliacao**. Ela utiliza alocação dinâmica de memória para expandir o tamanho do array conforme necessário.
- **void adicionar_avaliacoes(Aluno *aluno, Avaliacao avaliacao);**
 - A função void **adicionar_avaliacoes(Aluno *aluno, Avaliacao avaliacao)** tem como objetivo adicionar uma nova avaliação à lista de avaliações de um aluno. Essa função manipula a estrutura de dados do aluno, garantindo que a nova avaliação seja armazenada corretamente, utilizando alocação dinâmica de memória realizada pela função **realocar_memoria_avaliacao**.
- **void liberar_memoria(Aluno *alunos, int num_alunos);**
 - A função void **liberar_memoria(Aluno *alunos, int num_alunos)** tem como objetivo liberar a memória alocada dinamicamente para um array de estruturas do tipo **Aluno** e quaisquer recursos associados a cada aluno, evitando vazamentos de memória.

- `void calcular_notas(Aluno *aluno);`

- A função `void calcular_notas(Aluno *aluno)` tem como objetivo calcular a nota final de um aluno, ajustando as notas fornecidas de acordo com as regras de validação e, em seguida, realizando o cálculo da nota final com base nas notas ajustadas.

- 1. **Validação das notas:**

- Se uma nota for menor que 0, ela deverá ser ajustada para 0.
 - Se uma nota for maior que 10, ela deverá ser ajustada para 10.

- 2. **Cálculo da nota final:**

- A nota final será calculada com base nas notas ajustadas. O cálculo da nota final deverá seguir a seguinte fórmula.

$$NF = \frac{AV1 + AV2}{2}, \quad \text{onde } 0 \leq NF \leq 10$$

$$AV1 = \left(\frac{AP1 + AP2 + AP3}{3} \times 0.3 \right) + (NP \times 0.7)$$

$$AV2 = \left(\frac{AP1 + AP2 + AP3}{3} \times 0.3 \right) + (NP \times 0.7)$$

Onde:

- (AP1): Atividade Prática 1
 - (AP2): Atividade Prática 2
 - (AP3): Atividade Prática 3
 - (NP): Nota da Prova

Compilação do Código

```
gcc -o main main.c
```

Execução do Código

```
./main notas.csv
```

Critérios de Avaliação

- A avaliação considerará os seguintes aspectos:
 - Compilação e execução do programa (**2 pontos**).
 - Utilização adequada de alocação dinâmica e limpeza de memória (**3 pontos**).
 - Tratamento das notas de entrada (**2 pontos**).
 - Cálculo correto da nota final (**1 ponto**).
 - Organização, legibilidade e comentários no código (**2 pontos**).

Uso de IA e consulta a códigos da *Internet*

- [] Declaro que os códigos fornecidos por mim para resolver esta avaliação são de minha autoria. Não utilizei inteligência artificial, códigos da *Internet* ou ferramentas correlatas para resolver os problemas apresentados.
- [] Declaro que os códigos fornecidos por mim para resolver esta avaliação não são inteiramente de minha autoria, pois utilizei inteligência artificial, códigos da *Internet* ou ferramentas correlatas para ajudar a resolver os problemas apresentados. Estou ciente de que o uso de tais recursos resultará em um decréscimo de 50% na minha nota final.