UNIVERSIDADE DO ESTADO DE MINAS GERAIS
UEMG

CURSO: Engenharia da Computação **TURMA: 1º Período**
DISCIPLINA: Algoritmos e Estrutura de Dados 1
Professor: Willyan Michel Ferreira **Nota**:___/ **20**

Aluno(a): _____

1. **Sistema de Gestão de Funcionários com Vetor Dinâmico:** Desenvolva um programa em C para gerenciar uma lista de funcionários de uma empresa com as seguintes funcionalidades: **(12,5 pontos)**

   - **Cadastro de Funcionários:** O programa deve permitir o cadastro de funcionários usando um vetor alocado dinamicamente. Cada funcionário deve ter os seguintes dados (utilize uma struct para representar o funcionário):
     - Nome (máximo 50 caracteres)
     - ID (inteiro, deve ser único)
     - Salário (float)
     - Data de Contratação (dia, mês e ano)

   - **Atualização de Dados:** O programa deve permitir a atualização dos dados de um funcionário (nome, salário e data de contratação) com base no ID do funcionário.

   - **Exclusão de Funcionário:** O programa deve permitir a exclusão de um funcionário pelo ID. Após a exclusão, o vetor deve ser atualizado para não deixar lacunas.

   - **Gravação de Dados em Arquivo:** O programa deve possuir uma função que grave todos os dados dos funcionários cadastrados em um arquivo de texto chamado funcionarios.txt. Cada linha do arquivo deve conter os dados de um funcionário no seguinte formato:

     ID Nome Salário Dia/Mês/Ano

   - **Leitura de Dados do Arquivo:** O programa deve possuir uma função que leia os dados de um arquivo de texto funcionarios.txt e preencha um vetor alocado dinamicamente com esses dados. Ao final da leitura, o programa deve exibir na tela todos os funcionários lidos.

   - **Busca de Funcionário por ID:** O programa deve permitir que o usuário busque um funcionário pelo número do ID. Se o funcionário for encontrado, o programa deve exibir os dados dele. Caso contrário, exibir uma mensagem informando que o ID não foi encontrado.

   - **Liberação de Memória:** Após a leitura dos dados ou o cadastro dos funcionários, o programa deve liberar toda a memória alocada dinamicamente para o vetor de funcionários.

```
Sistema de Gestão de Funcionários
1. Cadastrar Funcionário
2. Atualizar Dados do Funcionário
3. Excluir Funcionário
4. Gravar Dados em Arquivo
5. Ler Dados do Arquivo
6. Buscar Funcionário por ID
7. Sair
Escolha uma opção: 2

Atualização de Dados do Funcionário:
Digite o ID do funcionário: 101
Digite o novo nome do funcionário: Alice Smith
Digite o novo salário do funcionário: 4600.75
Digite a nova data de contratação (dia mês ano): 16 3 2023
```

Exemplo Manipulação Exercício 1

2. Transformação e Manipulação de Vetores com Estruturas:

Desenvolva um programa em C que implemente um sistema para gerenciar um vetor de registros de estudantes. O programa deve incluir as seguintes funcionalidades: **(7,5 pontos)**

- **Leitura do Vetor de Estudantes:** Solicite ao usuário o número de estudantes (N). Alocar dinamicamente um vetor para armazenar as informações dos estudantes. Para cada estudante, leia os seguintes dados:
  - Nome: uma string com até 49 caracteres.
  - Matrícula: um número inteiro.
  - Nota: um valor float.
  -
- **Transformação dos Dados:** Implemente a função
  void transformarNotas(Estudante *vetor, int n)
  , onde:
  - Estudante é uma estrutura que contém os campos nome, matricula e nota.
  - A função deve transformar a nota de cada estudante para a média das notas dos estudantes vizinhos:
    - ✓ Para o primeiro estudante, considere o vizinho anterior como zero.
    - ✓ Para o último estudante, considere o vizinho próximo como zero.
    - ✓ Para os demais estudantes, calcule a média das notas do estudante anterior e do estudante próximo.

- **Exibição dos Dados:** Exiba o vetor de estudantes com as notas transformadas, mostrando o nome, matrícula e a nova nota para cada estudante.

- **Gravação em Arquivo:** Grave os dados dos estudantes transformados em um arquivo de texto chamado Resultado.txt. O arquivo deve ter uma linha para cada estudante, com o formato: Nome Matrícula Nota.

- **Liberação de Memória:** Após a utilização do vetor, libere a memória alocada dinamicamente para o vetor de estudantes.

REQUISITOS

Estrutura de Dados: Use uma estrutura chamada Estudante para armazenar as informações dos estudantes.

Alocação Dinâmica: Utilize alocação dinâmica de memória para o vetor de estudantes.

FUNÇÕES

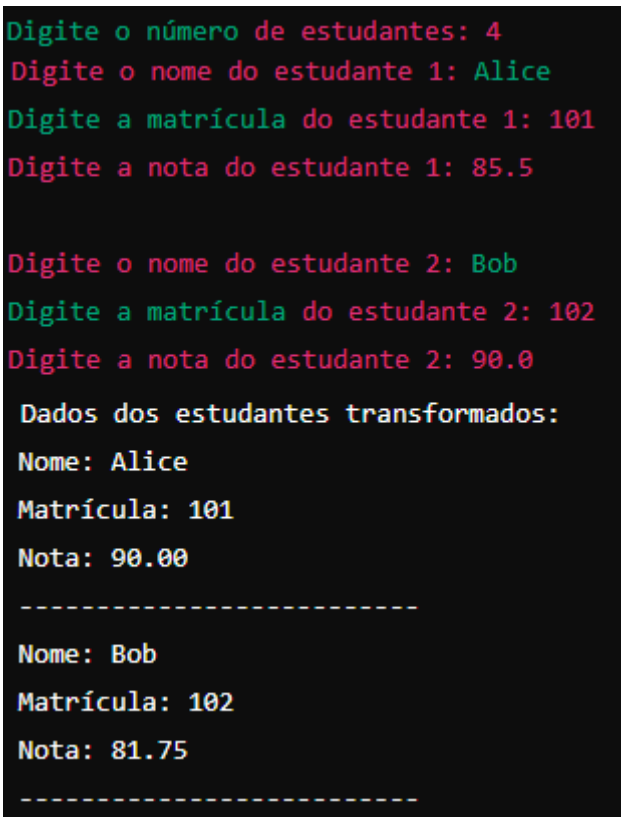
void lerEstudantes(Estudante *vetor, int *n): Para ler os dados dos estudantes e alocar memória.

void transformarNotas(Estudante *vetor, int n): Para transformar as notas dos estudantes conforme descrito.

void exibirEstudantes(Estudante *vetor, int n): Para exibir os dados dos estudantes.

void gravarEmArquivo(Estudante *vetor, int n, const char *nome_arquivo): Para gravar os dados dos estudantes em um arquivo de texto.

void liberarMemoria(Estudante *vetor): Para liberar a memória alocada.



Exemplo Manipulação Exercício 2