

# Trabalho 1

Este trabalho vale 3,0 pontos e é parte da nota da avaliação da Unidade 1. O trabalho poderá ser realizado em grupos de até 3 pessoas. Os arquivos deverão ser entregues compactados via SIGAA na tarefa designada para isso. Juntamente com os arquivos do TAD deverá ser incluído um arquivo com o **Relatório de atividades** desempenhadas pelo grupo. No relatório deverá constar a identificação dos autores, e o relato de como foi feita a divisão de tarefas, as dificuldades enfrentadas pelo grupo e como a atividade contribuiu para o aprendizado do conteúdo.

1. Faça um TAD para representar vetores de uma dimensão para valores reais alocados dinamicamente, o tamanho do vetor é definido em tempo de execução. Para isso será definido um tipo abstrato, denominado **Vetor**, e o conjunto de funções que operam sobre esse tipo. O tipo **Vetor** deverá fornecer as seguintes operações:

**cria** Função que cria um vetor de tamanho **n**.

```
Vetor* criaVetor(int n);
```

**libera** Função que libera a memória alocada para um vetor.

```
void liberaVetor(Vetor* v)
```

**tamanho** Função que retorna o tamanho de um vetor.

```
int tamanhoVetor(Vetor* v)
```

**acessa** Função que acessa o conteúdo de uma determinada posição **i** do vetor.

```
float acessa(Vetor* v, int i);
```

**atribui** Função que atribui um conteúdo **c** a uma determinada posição **i** do vetor.

```
void atribui(Vetor* v, int i, float c);
```

**maior** Função que retorna o maior elementos do vetor.

```
float maior(Vetor* v);
```

**menor** Função que retorna o menor elementos do vetor.

```
float menor(Vetor* v);
```

**listar** Função que lista todos os elementos do vetor. A listagem deve se apresentar de forma organizada como uma lista na mesma linha.

```
void listarVetor(Vetor* v);
```

**somar vetor** Função que retorna a soma de todos os elementos de um vetor.

```
float somaVetor(Vetor* v);
```

**somar elementos** Função que retorna a soma dos **n** primeiros elementos do vetor.

```
float somaElementos(Vetor* v, int n);
```

**somar vetores** Função que retorna um vetor resultante da soma dos elementos correspondentes de dois vetores. A função deverá verificar se os vetores passados tem o mesmo tamanho.

```
Vetor* somaVetores(Vetor* v1, Vetor* v2);
```

**mostrar soma vetores** Função que mostra a soma dos elementos correspondentes de dois vetores do mesmo tamanho, como mostrado no exemplo abaixo.

entrada

v1 = {1.0, 4.5, 7.0}

v2 = {6.5, 10.0, 8.5}

saída

1.0 + 6.5 = 7.5

4.5 + 10.0 = 14.5

7.0 + 8.5 = 15.5

A função deverá verificar se os vetores passados tem o mesmo tamanho.

```
void somaVetores(Vetor* v1, Vetor* v2);
```

2. O tipo estruturado deverá ser definido da seguinte forma:

```
struct vetor {  
    int tamanho;  
    float *elementos;  
};
```

3. No arquivo de cabeçalho deverá ter a redefinição do tipo, usando **typedef**, para se chamar **Vetor**, além da disponibilização de todas as funções que devem ser visíveis para o usuário do TAD. Funções que são apenas auxiliares para as funções principais não devem ser visíveis para o usuário. Todas as funções devem vir acompanhadas de comentários para compor a documentação, indicando a funcionalidade da operação. No topo do arquivo deverá ter uma descrição geral do TAD, indicando também os autores como comentário.
4. Todas as funções deverão fazer validações, quando necessárias, como verificação de entradas de tamanho e índices para evitar uso de tamanhos negativos, ou acessos e atribuições à índices fora da faixa do vetor. Nestes casos deve ser mostrado uma mensagem de erro.

#### Observações importantes:

- No arquivo entregue deverá constar o arquivo de cabeçalho, o arquivo de implementação e o Relatório de atividades.
- O protótipo das funções e a definição da estrutura deverá ser obedecido como indicado anteriormente.
- Não serão aceitas cópias literais de trabalhos.
- Não serão aceitos trabalhos entregues via e-mail, somente pela tarefa do SIGAA dentro do prazo estabelecido que será dia 28/10/2023 às 23:59.