

# MC202AB - Estrutura de Dados

## Lab 3 - Tipo Abstrato de Dados

**Data da Primeira Chance:** 12 de setembro de 2022

**Peso:** 2

Após a tragédia de Mariana, aumentou-se a atenção sobre o risco de rompimento de várias represas (barragens) de detritos de minérios pelo Brasil. Além disso, as grandes represas de água do país também intensificaram sua fiscalização.

Você é um engenheiro contratado pelo governo federal brasileiro para fiscalizar diariamente o nível e a segurança de diversas represas por todo o país. Você recebe os dados com todas as medições de nível feitas ao longo daquele dia para cada uma das represas pelas quais você é responsável.

Os dados encaminhados a você contém o nome da represa (com tamanho máximo de 100 caracteres), o valor **M** que representa a capacidade máxima dela, o valor **N** ( $4 \leq N \leq 1000$ ) que representa a quantidade de valores medidos, seguido de **N** medidas (em  $\text{km}^3$ ) de níveis atuais da represa. Exemplo:

```
Represa de ilha Solteira
21.20 5 20 20 20 20 20
```

Se houver qualquer incidente, você é o engenheiro responsável. Por isso, você deve emitir um único **alerta textual** caso alguma medida da represa atinga 90% de sua capacidade máxima. A mensagem do alerta deverá ser: “CAPACIDADE ACIMA DE 90% DA MÁXIMA!”.

Um problema adicional do seu trabalho como engenheiro é que sistema foi invadido e os nomes de represa foram todos modificados, dificultando a análise de saída da sua tarefa. Para isso, você deve padronizar os nomes de represa da seguinte forma (não necessariamente nesta ordem):

- Somente letras minúsculas
- Caracteres de *underline* (“\_”) devem ser substituídos por *espaço* (“ ”)
- Somente o nome da represa sem o prefixo “represa de ” (Ex.: “represa\_de\_itaipu” se tornaria “itaipu”)
  - “represa\_de\_” é o único prefixo possível
- Caso haja números no nome, remova (Ex.: “represa\_de\_ilha\_solteira3” se torna “ilha solteira”)
  - Por sorte, o invasor fez com que estes números poderão estar apenas no fim do nome da ilha...

**Obs.:** Os nomes das represas poderão se repetir na entrada, com diferentes medidas. Por exemplo, se a mesma represa aparece 3 vezes na entrada, ela deve ter 3 relatórios de saída na mesma ordem da entrada.

Para fazer essa fiscalização, crie um TAD para o tipo “*Represa*” contendo os seguintes itens:

- Nome da represa
- Valor decimal (*float*) do nível máximo da represa
- Um valor inteiro que representa a quantidade de medidas existentes no vetor de níveis
- Vetor de valores inteiros contendo as medidas de nível
- Valor inteiro que armazena o nível médio (média aritmética das medidas do vetor acima) da represa

Além disso, as seguintes operações devem ser declaradas para esse TAD:

- Uma função que lê dados de uma represa, inseridos na entrada do programa na forma

```
[nome]
[M] [N] [medida 1] ... [medida N]
```

e retorna um dado do tipo *Represa*;

- Note que a o campo que armazena o valor da média neste dado *Represa* precisa ser calculado para então ser inserido
- Uma função que recebe um dado do tipo *Represa* e retorna 1 caso alguma das medidas desta *Represa* tem valor maior ou igual do que 90% de sua capacidade máxima, ou retorne 0 caso contrário;
- Uma função que recebe um dado do tipo *Represa* e imprime na tela informações deste dado na forma:

```
NOME: [nome]
QTD NÍVEIS: [quantidade de medidas lidas]
MÉDIA: [média das medidas]
[alerta textual (se houver)]
#####
```

- Lembrando que a média dos níveis lidos deve ser um valor inteiro;
- O alerta textual só deve ser mostrado caso alguma das medidas de nível de represa seja maior que 90%
- Uma função que recebe como parâmetro um vetor de *Represa* seguido de um inteiro representando o tamanho do vetor e faça o pré-processamento dos nomes das represas. Detalhes importantes:
  - Essa função irá alterar o conteúdo de cada *Represa* do vetor, ou seja, não é necessário criar novas variáveis do tipo *Represa* e gerar um novo vetor;

- Essa função deve chamar outras funções que você deve definir para tratar estas strings.

É necessário seguir os conceitos e ideias de TAD que você aprendeu em MC202, usando *Cliente* → *Interface* → *Implementação* para esta tarefa de fiscalização. Então deverão ser criados três arquivos: **represa.h**, **represa.c** e **lab03.c** representando a interface, a implementação e o cliente, respectivamente.

**Dica:** Use as funções a seu favor. Declare e use as funções para as operações listadas acima, mas também declare e use funções que ache necessário para sua implementação.

## Entrada

Como dito anteriormente, a tarefa é fazer leitura e análise de dados de medição de uma represa. Então será dado como entrada os dados de **T** represas seguindo a ordem:

- Valor **T** ( $4 \leq T \leq 1000$ ) representando a quantidade de represas a serem analisadas
- **T** linhas de dados de represa, em que cada linha segue o formato

Portanto, o formato da entrada será:

```
[T]
[nome]
[M] [N] [medida 1] ... [medida N]
[nome]
[M] [N] [medida 1] ... [medida N]
...
[nome]
[M] [N] [medida 1] ... [medida N]
```

## Saída

A saída desta tarefa é a saída da operação de “Listar represas” definida no enunciado, que representa o relatório de cada uma das **T** represas analisadas. Cada relatório deve ser separado com uma linha contendo “#####” (5 caracteres “#”). O formato da saída será:

```
NOME: [nome]
QTD NÍVEIS: [quantidade de medidas lidas]
MÉDIA: [média das medidas]
[alerta textual (se houver)]
#####
```

# Exemplos

## Exemplo 1:

### Entrada

```
3
represa_de_Serra_da_MeSa3
54.40 5 34 44 51 21 44
represa_de_SOBRADINHO
34.10 7 26 22 28 20 20 25 26
Ilha_SOLTEIRA
21.20 6 20 20 20 20 20 20
```

### Saída

```
NOME: serra da mesa
QTD NÍVEIS: 5
MÉDIA: 38
CAPACIDADE ACIMA DE 90% DA MÁXIMA!
#####
NOME: sobradinho
QTD NÍVEIS: 7
MÉDIA: 23
#####
NOME: ilha solteira
QTD NÍVEIS: 6
MÉDIA: 20
CAPACIDADE ACIMA DE 90% DA MÁXIMA!
#####
```

## Regras e Avaliação

Está proibido o uso das bibliotecas `<string.h>` e `<ctype.h>`.

Seu código será avaliado não apenas pelos testes do CodePost, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código iremos analisar: o uso apropriado de funções, de comentários, e de structs; a escolha de bons nomes de funções, variáveis e de structs e seus campos; o correto uso de Tipos Abstratos de Dados e a correta separação em vários arquivos; a ausência de diversos trechos de código repetidos, e o tempo de execução e uso de memória dos algoritmos projetados. Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

## Submissão

Você deverá submeter **três arquivos** na tarefa *Lab 3 - Tipo Abstrato de Dados* no CodePost: ***represa.h***, ***represa.c*** e ***lab03.c***. Após o prazo estabelecido para a atividade, será aberta uma tarefa Lab 3 - Segunda Chance, com prazo de entrega até o fim do semestre.