

## Engenharia e Ciência da Computação – Estruturas de Dados

### Trabalho 1 – Matrículas em Disciplinas

#### Informações Gerais

- **Data Limite de Entrega:** 15/10/2023 (23:59).
- **Pontuação:** 10 pontos (10% da nota do semestre).
- **Formato de Entrega:** Os arquivos produzidos no trabalho devem ser compactados em formato **.zip** e submetidos na tarefa do ambiente virtual de aprendizagem (AVA).
- Os trabalhos devem ser desenvolvidos individualmente.
- **Importante:** Trabalhos entregues após a data limite sem justificativa com comprovação documental (atestado médico, etc.), ou que não estiverem de acordo com o especificado receberão nota zero.

#### Contextualização

Considere um sistema acadêmico em que existe o registro de disciplinas, estudantes e a matrícula de estudantes em disciplinas. Cada **estudante** possui nome (string), número de matrícula (inteiro) e e-mail (string). Cada **disciplina** possui nome (string), código (string), professor (string), uma lista de matrículas (cada item da lista é um ponteiro para uma variável do tipo matrícula) e uma lista de pré-requisitos (cada item da lista é um ponteiro para outra disciplina). Cada **matrícula** possui um ponteiro indicando o aluno que está matriculado, a nota final do aluno (float, entre 0 e 10), o percentual de presença (float, entre 0 e 1) e um inteiro indicando se o aluno foi aprovado (1) ou não (0). O objetivo do trabalho é gerar alguns relatórios com base nestes dados.

#### Entradas e Saídas

As entradas do programa serão digitadas na entrada padrão e as saídas deverão ser mostradas na tela. A entrada consiste no nome de um arquivo contendo dados de estudantes, disciplinas e matrículas e um número inteiro indicando o relatório que deve ser exibido na tela.

Um exemplo de entrada seria:

```
entrada.txt  # <- arquivo de entrada que contém os dados
5           # <- relatório a ser produzido
```

O formato do arquivo de entrada é dado por:

```
<número de alunos>
<nome do aluno>;<matrícula>;<email>
<nome do aluno>;<matrícula>;<email>
<número de disciplinas>
<nome da disciplina>;<código>;<professor>
<nome da disciplina>;<código>;<professor>
<número de dados de pré-requisitos>
<código da disciplina>;<código do pré-requisito>
<código da disciplina>;<código do pré-requisito>
<número de matrículas>
<código da disciplina>;<matrícula do aluno>;<nota>;<presenças>;<aprovado: 0 ou 1>
```

<código da disciplina>;<matrícula do aluno>;<nota>;<presenças>;<aprovado: 0 ou 1>

Por exemplo:

```
3
Joao Jamelao;10;joao@email.com
Maria Miranda;3;maria@email.com
Lucas Louzada;888;lucas@email.com
2
Programacao I;INF-1;Thiara
Estruturas de Dados;INF-2;Filomena
1
INF-2;INF-1
4
INF-2;888;3;1;0
INF-2;888;7;0.1;0
INF-2;888;9;1;0
INF-1;3;10;0.9;1
INF-2;3;9;0.8;1
INF-2;10;7;0.93;1
INF-1;888;4;0.3;0
```

Os relatórios a serem implementados são descritos a seguir:

1. **Alunos matriculados:** leia o código de uma disciplina e mostre na tela os nomes dos alunos que possuem matrícula na disciplina, um por linha, e sem repetição.
2. **Pré-requisitos diretos:** leia o código de uma disciplina e mostre na tela os nomes das disciplinas que são pré-requisitos, um por linha. Os dados devem ser exibidos na ordem em que são dados no arquivo de entrada.
3. **Pré-requisitos completos:** leia o código de uma disciplina e mostre na tela os nomes da cadeia completa de pré-requisitos. Por exemplo, se B e C são pré-requisitos de D e A é um pré-requisito de B, devem aparecer na lista A, B e C. Os pré-requisitos diretos devem ser exibidos na ordem em que são dados no arquivo de entrada. Se B e C são pré-requisitos de D, os pré-requisitos de B devem ser exibidos após B e antes de C. Isto é, os pré-requisitos de uma disciplina devem ser exibidos após ela e antes da próxima que está no mesmo nível.
4. **Disciplinas matriculadas:** leia o número de matrícula de um aluno e mostre na tela a lista de disciplinas que o aluno possui matrícula, sem repetição.
5. **Reprovações por Professor:** para cada professor, mostre na tela o seu nome e o número de reprovações em disciplinas que ele ministrou. Apresente os dados de forma ordenada do maior número de reprovações para o menor.
6. **Aprovações por aluno:** leia o número de matrícula de um aluno e mostre na tela a lista de disciplinas em que ele foi aprovado (status = 1 ou (nota final  $\geq$  5 e percentual de aprovação  $\geq$  75%)). Mostre os dados de forma ordenada pelo nome da disciplina, em ordem alfabética.
7. **Status inconsistente:** por um erro no sistema, em algumas matrículas, o status de aprovação está inconsistente com os valores de nota final e percentual de presença. No arquivo de exemplo acima, na linha "INF-2;888;9;1;0", o aluno teve nota 9 e 100% de presença e foi marcado como reprovado (status = 0). Mostre na tela a lista de linhas inconsistentes no mesmo formato do arquivo.
8. **Pré-requisitos não cursados:** devido a um erro no sistema, alguns alunos foram matriculados em disciplinas sem ter cursado os pré-requisitos. No arquivo de exemplo acima, o aluno com

matrícula 10 cursou a disciplina INF-2 sem ter cursado primeiro INF-1. Mostre na tela o número de matrículas em que isto aconteceu. Note que se A é pré-requisito de B e B é pré-requisito de C, e o aluno cursou C, é necessário verificar se ele cursou tanto B quanto A.

9. **Pré-requisitos não aprovados:** devido a um erro no sistema, alguns alunos cursaram os pré-requisitos, mas não foram aprovados, e mesmo assim foram capazes de se matricular em uma disciplina. No arquivo de exemplo acima, o aluno 888 cursou a disciplina INF-2 sem primeiro ser aprovado em INF-1 (embora ele tenha cursado INF-1). Mostre na tela o número de matrículas em que esta inconsistência aconteceu. Como no caso anterior, se A é pré-requisito de B e B é pré-requisito de C, e o aluno cursou C, é necessário verificar se ele foi aprovado tanto em B quanto em A.
10. **Resumo dos Alunos:** Para cada aluno, mostre na tela o nome do aluno, o número de matrícula, o número de disciplinas em que ele foi aprovado, o número de disciplinas cursadas e a nota média nas disciplinas. Exiba os dados de forma ordenada pelo nome do aluno, em ordem alfabética.
11. **Resumo das disciplinas:** Para cada disciplina, mostre na tela o código da disciplina, seu nome, o nome do professor da disciplina, o número de matrículas, o número de aprovações e a nota média dos alunos. Exiba os dados de forma ordenada pelo percentual de aprovações, do maior para o menor.
12. **Remoção de disciplinas:** leia o código de uma disciplina e remova todas as menções à disciplina nas estruturas de dados (tanto na lista de disciplinas, quanto de matrículas e pré-requisitos). Mostre na tela qual seria o conteúdo do arquivo de entrada após a remoção da disciplina. Não é válido resolver a questão apenas usando condicionais para selecionar os dados a exibir na tela. A disciplina deve de fato ser removida das listas encadeadas.

### Regras

- No trabalho, só é permitido utilizar *arrays* (alocados estaticamente ou dinamicamente) para representar strings. As demais coleções de dados devem ser armazenadas apenas em listas encadeadas.
- Deve ser implementada uma lista encadeada genérica.
- As estruturas de dados **não** precisam ser opacas.
- As estruturas aluno, disciplina e matrícula devem possuir exatamente os campos especificados na seção de Contextualização.
- Outras estruturas de dados podem ser criadas se o aluno julgar necessário.
- Com exceção de ponteiros, dados não devem ser replicados entre estruturas de dados.
- Todos os floats devem ser exibidos com 2 casas depois da vírgula.
- Envio de códigos que não compilam ou erros no valgrind do tipo *invalid read* ou *invalid write* resultarão em nota zero.
- Número de alocações diferente do número de *freets* resultará em punições graves na nota.
- Códigos replicados quando poderiam ser criadas funções reutilizáveis ( $\geq 75\%$  de similaridade entre trechos de código) resultarão em punições graves na nota.