

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL

FACOM - FACULDADE DE COMPUTAÇÃO

ALGORITMOS E PROGRAMAÇÃO II

Trabalho Prático

Sistema de Gerenciamento de Alunos

1 DESCRIÇÃO

O trabalho prático de nossa disciplina consiste na implementação de um sistema para gerenciamento de alunos da disciplina de Empreendedorismo. Cada aluno da disciplina possui as seguintes informações armazenadas:

- Nome (cujos caracteres não devem incluir acentos e cedilhas);
- RA (inteiro);
- Nota da primeira prova obrigatória (P1);
- Nota da segunda prova obrigatória (P2);
- Nota do trabalho prático (T);
- Nota da prova optativa (PO).

Todas as notas devem ser representadas usando números em ponto flutuante (utilize tipo double). A média final **MF** é calculada de acordo com a seguinte fórmula:

$$MF = 0,35 * P1 + 0,35 * P2 + 0,3 * T,$$

lembrando que a nota da prova optativa (PO) **pode substituir** a menor das notas de prova (P1 ou P2), caso seja maior que tal nota. O aluno cuja média final for maior ou igual a **6,0** será considerado aprovado, caso contrário, estará reprovado.

Durante a execução do programa, deve ser utilizado um **vetor de registros ordenado** de alunos (a ordem deve ser a lexicográfica considerando os nomes dos alunos, caso existam dois ou mais alunos com o mesmo nome, resolva o "empate" colocando aquele que tem menor RA na frente). Sua aplicação deve fornecer um **menu** com as seguintes opções:

1. **Cadastrar um novo aluno:** insere um novo aluno na lista contendo os dados fornecidos como entrada pelo usuário. O usuário deve fornecer dados para todos os campos do aluno (se a PO não tiver sido realizada, a nota zero deve ser informada);

2. **Buscar aluno:** solicita ao usuário uma cadeia de caracteres e lista os nomes completos, a média final e a situação (**aprovado** ou **reprovado**) de todos os alunos cujos nomes casam (total ou parcialmente com a cadeia fornecida). A listagem deve mostrar ao final uma contagem do número de resultados encontrados;
3. **Ler arquivo com listagem de alunos:** solicita ao usuário o nome de um arquivo de texto contendo os dados de alunos. Esse arquivo conterá os dados de, no máximo, um aluno por linha e, em cada linha, **todos** os dados do aluno deverão ser informados na mesma ordem descrita na descrição da estrutura que define um aluno. Se o aluno não fez PO, o arquivo deve conter a nota zero no local adequado. Ao final do cadastramento, deve-se mostrar **na tela** uma contagem de quantos alunos foram cadastrados. Veja a Seção 3 para mais informações sobre o formato do arquivo;
4. **Gerar arquivo com dados dos alunos aprovados, ordenados em ordem alfabética:** escreve um arquivo texto contendo os nomes, RAs e as médias finais dos alunos aprovados na disciplina, um aluno por linha. Os alunos constantes no arquivo resultante devem estar em ordem alfabética de nome. (OBS: caso existam dois ou mais alunos com o mesmo nome, resolva o "empate" colocando aquele que tem menor RA na frente). Ao final do arquivo, deve aparecer a contagem de alunos aprovados. **O nome do arquivo de saída deve ser aprovados.txt.** Veja a Seção 4 para mais informações sobre o formato do arquivo;
5. **Gerar arquivo com dados dos alunos reprovados, ordenados em ordem alfabética:** escreve um arquivo texto contendo os nomes e as médias finais dos alunos reprovados na disciplina, um aluno por linha. Os alunos constantes no arquivo resultante devem estar em ordem alfabética de nome. Ao final do arquivo, deve aparecer a contagem de alunos reprovados. **O nome do arquivo de saída deve ser reprovados.txt.** Veja a Seção 4 para mais informações sobre o formato do arquivo;
6. **Fechar programa:** encerra a execução do programa. Antes do encerramento, salve todos os dados dos alunos armazenados no vetor em um arquivo de saída chamado "**atual.txt**", colocando um aluno por linha, com os campos separados por espaço e na ordem em que foram apresentados na Seção 1.

O algoritmo de ordenação a ser utilizado deverá ser um dos seguintes algoritmos vistos em sala de aula: ordenação por inserção, ordenação por seleção.

2 EXECUÇÃO

A execução do programa deve se iniciar mostrando o menu de opções. O usuário pode escolher a qualquer momento uma das opções e, baseado na escolha realizada, o sistema deve realizar a atividade pedida.

Os cadastros de novos alunos, tanto individualmente quanto por arquivo, devem fornecer dados para todos os campos do(s) aluno(s) em questão.

3 FORMATO DO ARQUIVO DE ENTRADA

Os arquivos de entrada para as opções que envolvem leitura de arquivo possuem **uma linha para cada aluno** a ser cadastrado. Cada linha possui dados para **todos os campos** do tipo de aluno em questão e é encerrada com um caractere de nova linha. Os campos que devem ser considerados são APENAS os descritos na Seção 1 e na MESMA ordem da descrição.

4 FORMATO DO ARQUIVO DE SAÍDA

Os arquivos de saída para as opções que envolvem escrita em arquivo devem possuir **uma linha por aluno**. Cada linha possui os dados dos alunos conforme mencionado nas opções de menu 4 e 5. Os campos que devem ser escritos são APENAS os descritos na Seção 1 e na MESMA ordem da descrição. Ao final deve aparecer a linha com a contagem seguindo o formato:

Total: cont

na qual **cont** é o número de alunos aprovados ou reprovados, de acordo com a opção escolhida.

Ao final de cada linha, deve ser colocado o caractere de nova linha.

5 ESPECIFICAÇÕES

- O trabalho prático poderá ser realizado em grupos de, no máximo, 2 alunos, sem **exceções**.
- O trabalho deverá ser implementado usando a linguagem C/C++. Você deverá criar o seu programa usando dois ou mais arquivos .CPP (com o auxílio de arquivos .h), sendo que um deles deverá conter apenas a função main e bibliotecas, e no(s) outro(s) arquivo(s) todas as demais funções utilizadas pelo

programa.

- No momento de teste dos programas, os arquivos de entrada serão armazenados dentro da mesma pasta dos arquivos fonte, mesmo local onde os de saída devem ser criados.
- Poderão ser realizadas entrevistas com os componentes dos grupos.
- Trabalhos copiados receberão nota zero.
- A entrega de todos os códigos implementados deve ser realizada até o dia **15/11/2021** em um arquivo compactado (contendo também um README com os nomes dos integrantes do grupo) no link que estará disponível no AVA

6 AVALIAÇÃO

Seu trabalho será avaliado de acordo com a correção na implementação das **estruturas** e **funções** criadas, funcionamento correto dos algoritmos implementados, formatos de entrada e saída conforme descritos.

Programas que não compilarem ou derem violação de memória durante a execução receberão nota zero. Uma saída somente será considerada correta se for **exatamente** igual à esperada, caso contrário, a pontuação da funcionalidade em questão será zerada.

Os arquivos de entrada e de saída devem seguir **exatamente** os formatos especificados, caso isso não ocorra, as funcionalidades envolvidas terão sua pontuação zerada.