Universidade Federal do Rio Grande do Norte
Escola de Ciências e Tecnologia
ECT2303 – Linguagem de Programação 2017.2
Prof. Sandro Lopes

Projeto final - Sistema de Gerenciamento de Alunos de Linguagem de Programação

Leia com bastante atenção as instruções pedidas, para que você possa implementar corretamente o que se pede.

Você foi incumbido de popular para desenvolver um sistema simples de gerenciamento das atividades de uma turma de Linguagem de Programação. Neste sistema, devem existem operações convencionais de busca, inserção e remoção de alunos. Além disto, o sistema deve ser capaz de fornecer dados estatísticos sobre as notas dos alunos e a quantidade de faltas de cada um deles. Também, devem ser dadas informações sobre a relação dos alunos, para auxiliar o professor na toma de decisões como a elaboração de grupos e verificação de "colas" em trabalhos e provas.

Para tanto foram estabelecidas algumas diretrizes sobre o projeto. Estas devem ser seguidas como solicitado.

1 Documentação Entregue e Critérios de Avaliação

O documento a ser entregue consiste do código fonte do programa desenvolvido na linguagem C++. O aluno é encorajado a implementar o trabalho em grupo de **no máximo três membros**. O grupo deve decidir que tarefa ficará sob responsabilidade de cada membro, de forma que toda a carga de trabalho seja igualmente dividida entre todos.

Os alunos devem apresentar ao professor o trabalho desenvolvido. Na apresentação, os membros de cada grupo serão indagados sobre as técnicas de programação utilizadas, como cada funcionalidade proposta foi resolvida, quais seriam possíveis soluções alternativas e como foram coordenadas as atividades em grupo. Todos estes critérios serão levados em consideração para a atribuição da nota do trabalho. Além disto, o projeto deve utilizar os conceitos utilizados em sala de aula, como:

- Funções;
- Matrizes;
- Strings;
- Tipos estruturados;
- Leitura e escrita de/em arquivos.

O código fonte do projeto deve ser enviado via SIGAA impreterivelmente até as 23:59 do dia 03/12/2017. Após isto ser feito, cada grupo deve agendar um horário entre os dias 04/12 e 06/12 para a apresentação do trabalho ao professor da disciplina.

Não serão tolerados programas copiados sob nenhuma hipótese: em caso de constatação de cópia, o trabalho de todos os alunos do grupo será anulado. Além disto, conhecimentos avançados de programação, como classes, só serão aceitos caso o aluno fale antecipadamente com o professor. Em caso contrário, o trabalho será anulado.

2 Sobre as estruturas de dados

- Deve haver uma estrutura para representar um aulo. Nela, deve conter informações sobre o seu nome (com no máximo 50 caracteres), um inteiro com sua matrícula, um vetor com quatro elementos, informando a nota do aluno em cada uma das unidades e na prova final (caso seja necessária), e um inteiro armazenando o número de faltas.

- Deve ser criada uma estrutura para manipular, durante a execução, todos os possíveis alunos da disciplina. Para tanto, um **vetor de alunos** deve ser utilizado para tal finalidade. É importante mencionar que o número de alunos pode aumentar durante a execução do problema, então a capacidade do vetor de alunos deve ser suficiente para abrigar todos os envolvidos. Assuma que a capacidade máxima da sala é de 100 alunos.

- Além disto, deverá existir uma estrutura para armazenar informações sobre as iterações entre os alunos. Neste caso, deverá ser utilizada uma **matriz de interação entre os alunos**, ou apenas **matriz de interação**. Esta deverá ser uma matriz quadrada, onde o número de linhas e o número de colunas serão iguais ao número de alunos da disciplina envolvidos. O índice utilizado para acessar um determinado aluno deverá ser o mesmo para acessar a linha e a coluna desta matriz. Nela, cada elemento M[i][j] ou M[j][i] da matriz irá armazenar um valor inteiro informando o *status* da relação entre os alunos i e j. Entre os valores possíveis de relação são:

  - 0: não se conhecem;
  - 1: se conhecem, mas não se relacionam muito;
  - 2: se relacionam bastante.

  Como esta é uma relação recíproca, considere que $M[i][j] = M[j][i]$.

- Tanto os dados dos alunos quanto os dados das interações entre eles deverão ser salvos em arquivos. O arquivo "alunos.txt" deverá se encarregar de manter o conjunto de alunos da turma, enquanto que o arquivo "relacao.txt" deve armazenar a matriz de interação dos alunos. Um exemplo da formatação do arquivo "alunos.txt" pode ser vista a seguir:

```
3
João da Silva
2013123456
6.5 1.2 4.5
4
José da Costa
2013500000
8.7 7.7 9.7
0
Maria Francisca
2013987654
1.1 1.4 1.6
36
```

  Na primeira linha, são informados o número de alunos presentes. A partir da segunda linha, cada sequencia de três linhas refere-se a um correntista específico. Na primeira linha, encontra-se o nome do aluno; na segunda linha, a matrícula do aluno; na terceira linha, a lista com as notas do aluno, em sequência e separados por um

espaço; na quarta e última linha, o número de faltas. Um exemplo da formatação do arquivo "relacao.txt" pode ser vista a seguir:

```
3 3
0 2 1
2 0 0
1 0 0
```

Na primeira linha, são informados o número de linhas e de colunas da matriz de interação/relação. A partir da segunda linha, são informados os elementos M[i][j] da matriz de interação. Cada linha é separada por uma quebre de linha, e cada elemento da linha é separado dos outros por um espaço.

3 Operações de Consulta, Inserção e Remoção de Alunos

Os dados dos alunos precisam ser acessados para que o usuário possa fazer as operações existentes, principalmente com relação a matrícula, **pois este valor deve ser gerado aleatoriamente durante a inserção de um novo aluno**. Além disto, o sistema deve ter a capacidade de inserir novos alunos e excluir alunos caso seja necessário.

No caso da consulta aos dados, devem existir dois tipos:

1. Consulta por matrícula: Ao final, o sistema deve informar os dados do aluno (nome matrícula, notas e faltas) especificado pela matrícula;

2. Consulta por nome: Ao final, o sistema deve informar os dados dos alunos (nome matrícula, notas e faltas) com o nome informado. É importante mencionar que esta função pode retornam mais de um valor, pois podem existir vários alunos com o mesmo nome;

DICA: Para a primeira função, crie uma função que verifica se existe algum aluno com a matricula no vetor de aluno. Caso existe, retorne o índice para o vetor, caso contrário, retorne -1. Esta função de busca serão extremamente importantes nas funções a seguir. Para a segunda função, um vetor de índices deve ser disponibilizado com parâmetro de saída, bem como o número de elementos *por referência*. Caso um aluno com o nome passado como parâmetro seja encontrado, o índice deste deve ser inserido no vetor de índices, e o número de elementos deve ser atualizado.

Para a inserção de um novo aluno, o sistema deve receber, primeiramente, a matrícula do aluno. Depois, ele deve verificar se já existe no sistema algum aluno com a matrícula dada. Caso esteja, ele deve informar que não é possível fazer a inserção. Em caso contrário, ele deve inserir o nome, as notas dos alunos com valores nulos e o número de faltas com valor nulo. Em caso de sucesso, uma mensagem de êxito deve ser exibida para o usuário. Também, uma nova linha e uma nova coluna na matriz de interação deve ser estabelecida, cujas relações devem ser iniciadas em zero.

Para a exclusão de um aluno, deve ser passado apenas a matrícula. Caso não exista aluno com a matrícula dada, uma mensagem deve ser emitida informando que a operação não pode ser executada. Caso contrário, a exclusão deve ser realizada, e uma mensagem de êxito deve ser exibida para o usuário. Também, a linha e a coluna referente ao aluno deve ser excluída da matriz de interação.

4 Operações de Atualização de Dados dos Alunos

As operações de manipulação de dados disponibilizadas no sistema são para a inserção de notas, faltas e atribuição de relação entre os alunos. Nos três caso, o usuário deve informar apenas a(s) matrículas(s) dos alunos envolvido(s) para consulta.

Na inserção da nota de um aluno, primeiramente deve ser informada a matrícula. Depois, deve se verificar se existe algum aluno com a matrícula dada. Caso não exista, uma mensagem de erro deve ser enviada, informando que o aluno não existe. Em caso contrário, solicita-se a inserção da unidade e da nota para aquela unidade. Considere, neste caso, que a primeira unidade tem índice 1, a segunda unidade tem índice 2, a terceira unidade tem índice 3. O programa deve substituir a nota no campo assinalado e emitir uma mensagem de sucesso para o usuário.

Na inserção das faltas de um aluno, primeiramente deve ser informada a matrícula. Depois, deve se verificar se existe algum aluno com a matrícula dada. Caso não exista, uma mensagem de erro deve ser enviada, informando que o aluno não existe. Em caso contrário, solicita-se a inserção do número de faltas a ser *acrescido* (valor positivo) ou deduzido (valor negativo). O programa deve verificar se, após a operação, o número de faltas passa a ser menor do que zero; em caso positivo, o número de faltas deve ser reduzido a zero. Por fim, o programa deve emitir uma mensagem de sucesso para o usuário.

Na atribuição de relações entre alunos, primeiramente deve ser informada as matrículas dos dois alunos. A seguir, o programa deve verificar se as matrículas são iguais. Em caso verdadeiro, uma mensagem de erro deve ser informada ao usuário. Em caso falso, o programa passa a verificar se cada cada uma das matrículas define um aluno no conjunto (as duas matrículas precisam ser válidas). Em caso falso, uma mensagem de erro deve ser informada ao usuário. Em caso verdadeiro, o programa deve receber o *status* de relação entre os alunos e verificar se ele é um valor válido. Caso seja, substitui-se este valor nas posições M[i][j] e M[j][i] da matiz de interações. Em caso contrário, uma mensagem de erro deve ser enviada ao usuário.

5 Operações de Leitura e Escrita dos Arquivos

As operações de leitura e escrita dos arquivos se limitam a abrir e salvar dados nos arquivos "alunos.txt", e "relacao.txt". Em ambas as operações, a disposição das informações deve obedecer ao formato estipulado para cada arquivo. **É importante também garantir que no início da execução do sistema, as funções de leitura dos arquivos deve ser informadas, e ao final da execução do sistema, os dados dos clientes e das movimentações deve ser salvo nos arquivos respectivos.** Isto irá permitir ao programa iniciar com a configuração definida no final da última execução.

6 Operações para Análise Estatística do Vetor de Alunos

As operações estatísticas do sistema envolvem a análise das notas e das faltas dos alunos. Com relação ao vetor de alunos, duas operações serão realizadas. A primeira é o cálculo da média de faltas da turma. A segunda é a construção de um histograma de frequência a partir da média das notas dos alunos.

Para a obtenção da média de faltas da turma, basta somar os valores de frequência de todos os alunos e dividir pelo número total de alunos. O resultado deve ser exibido ao usuário.

Para o operação de exibição de um histograma de frequência a partir da média das notas dos alunos, considera-se com conjunto com 9 classes, cada classe sendo um intervalo de valores. Assume-se que as notas variam entre 0 e 10, de forma que as classes sejam $[0, 1[; [1, 2[; [2, 3[, \cdots, [9; 10]$. Com esta informação, você irá precisar criar funções auxiliares para verificar o maior e o menor valor média de notas. Com estes valores você poderá dividir os alunos em faixas de valores e depois contabilizar quantos correntistas estão em cada faixa. Ao final, um gráfico deverá ser exibido.

7 Operações para Obtenção de Dados da Matriz de Interações

No caso da matriz de interações, é possível analisar os dados referentes a um único aluno. Neste caso, as operações referente são:

- Listar dos alunos que estão relacionados ou não com um aluno;
- Verificar qual a relação entre dois alunos.

No caso de listar dos alunos que estão relacionados ou não com um aluno, primeiramente o usuário deve informar a matrícula do aluno em questão. A seguir, o programa deve verificar se existe algum aluno no vetor com a matrícula dada. Em caso negativo, uma mensagem de erro deve ser exibida. Em caso positivo, o usuário deve informar o *status* de relação, e o programa deve listar os alunos de apresentam aquele nível de relação com a aluno.

No caso de verificar qual a relação entre dois alunos, deve ser informada as matrículas dos dois alunos. A seguir, o programa deve verificar se as matrículas são iguais. Em caso verdadeiro, uma mensagem de erro deve ser informada ao usuário. Em caso falso, o programa passa a verificar se cada cada uma das matrículas define um aluno no conjunto (as duas matrículas precisam ser válidas). Em caso falso, uma mensagem de erro deve ser informada ao usuário. Em caso verdadeiro, o programa irá verificar o campo M[i][j] e imprimir a seguinte mensagem, de acordo com o *status*.

- Valor 0: "Não se conhecem";
- Valor 1: "Conhecem-se de vista";
- Valor 2: "São amigos".

8 Programa principal

No programa principal, deve ser disponibilizado um menu com todas as opções descritas anteriormente para o sistema, enquanto um valor de saída não for digitado pelo usuário. Os valores associados a cada operação e a saída do sistema podem ser atribuídos por você, devendo ser explicitamente indicados para o usuário do sistema. Um resumo das funcionalidades exigidas para o problema são dados a seguir:

1. Consulta por matrícula;
2. Consulta por nome;
3. Inserção de um novo aluno;
4. Remoção de um aluno existente;
5. Inserção de uma nota;
6. Inserção de uma falta;
7. Atribuição de relação entre alunos;
8. Média das faltas dos alunos;
9. Histograma das média das notas dos alunos;
10. Listagem dos alunos que estão relacionados ou não com um aluno;
11. Relação entre dois alunos.

A leitura dos arquivos “alunos.txt” e “relacao.txt” deve ser feita na inicialização do sistema, ou seja, no início do programa principal. Também, ao final da execução do programa, os dados disponibilizados devem ser armazenados nos arquivos “alunos.txt” e “relacao.txt”.