

## Trabalho para o 4º Bimestre

### Instruções

1. Este trabalho vale **50 pontos para a 4ª Avaliação**.
2. Este trabalho deve ser realizado por equipes de até três (03) alunos.
3. Em nenhuma hipótese serão aceitas equipes com mais de três alunos.
4. As equipes têm até **23h50m** do dia **28/11/2019**, para submeter os trabalhos no sistema Moodle. Após esta data e hora limite o sistema não aceitará mais submissões.
5. Os trabalhos deverão ser enviados compactados em formato .zip ou .rar, nomeados com os nomes dos integrantes da equipe.
6. Cada programa implementado deverá conter um comentário com a identificação da equipe.
7. Não será aceita entrega de trabalho por outros meios além do especificado nestas instruções.
8. Caso ocorram casos de plágio, as equipes envolvidas receberão nota zero.

---

#### Questão 1 (**20 pontos**): Matrizes esparsas

Considere o trabalho anterior sobre Matrizes esparsas:

Matrizes esparsas são matrizes nas quais a maioria das posições é preenchida por zeros. Para essas matrizes, podemos economizar um espaço significativo de memória se apenas os termos diferentes de zero forem armazenados. As operações usuais sobre essas matrizes (somar, multiplicar, inverter, pivotar) também podem ser feitas em tempo muito menor se não armazenarmos as posições que contêm zeros.

Uma maneira eficiente de representar estruturas com tamanho variável e/ou desconhecido é com o emprego de alocação encadeada, utilizando listas. Vamos usar essa representação para armazenar as matrizes esparsas. Cada coluna da matriz será representada por uma lista linear circular com uma célula

cabeça. Da mesma maneira, cada linha da matriz também será representada por uma lista linear circular com uma célula cabeça. Cada célula da estrutura, além das células cabeça, representará os termos diferentes de zero da matriz e deverá ser como no código abaixo:

```
typedef struct Celula
{
    Celula direita, abaixo;
    int linha, coluna;
    double valor;
} Celula;
```

O campo abaixo deve ser usado para referenciar o elemento diferente de zero na mesma coluna. O campo direita deve ser usado para referenciar o próximo elemento diferente de zero na mesma linha. Dada uma matriz A, para um valor A(i,j) diferente de zero, deverá haver uma célula com o campo valor contendo A(i,j), o campo linha contendo i e o campo coluna contendo j. Essa célula deverá pertencer a lista circular da linha i e também deverá pertencer à lista circular da coluna j. Ou seja, cada célula pertencerá a duas listas ao mesmo tempo. Para diferenciar as células cabeça, coloque -1 nos campos linha e coluna dessas células.

Considere a seguinte matriz esparsa:

$$A = \begin{pmatrix} 50 & 0 & 0 & 0 \\ 10 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 \\ -30 & 0 & -60 & 5 \end{pmatrix}$$

A representação da matriz A pode ser vista na Figura 1. Com essa representação, uma matriz esparsa m x n com r elementos diferentes de zero gastará ( m + n + r ) células. É bem verdade que cada célula ocupa vários bytes na memória; no entanto, o total de memória usado será menor do que as m x n posições necessárias para representar a matriz toda, desde que r seja suficientemente pequeno.

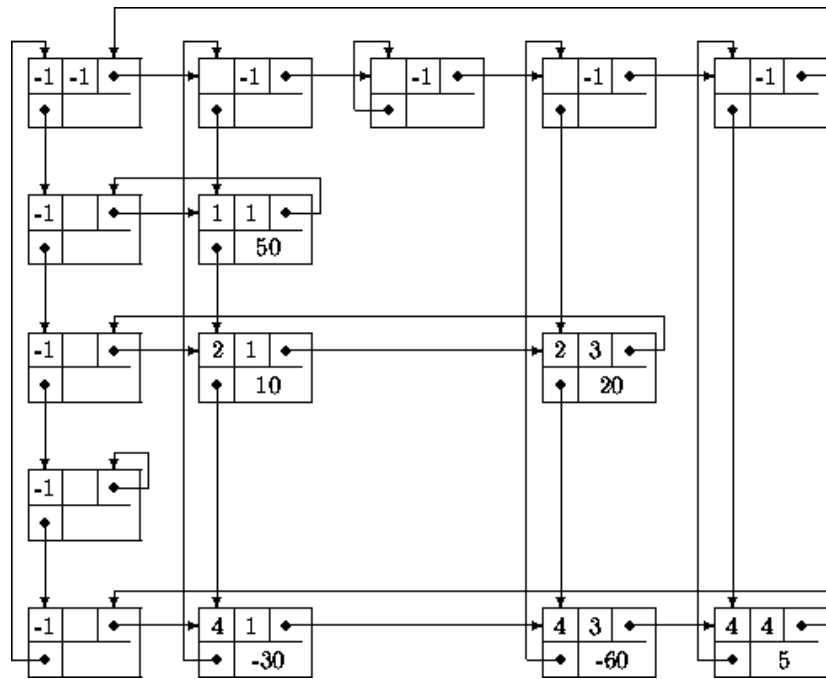


Figura 1: Representação de matriz esparsa como listas encadeadas.

Dada a representação de listas duplamente encadeadas, o problema consiste em desenvolver em linguagem C um tipo de dados Matriz com as seguintes operações, conforme esta especificação:

- void imprimeMatriz(): Essa função imprime (uma linha da matriz por linha na saída) a matriz A, inclusive os elementos iguais a zero.
- void leMatriz(): Essa função lê, do dispositivo de entrada, os elementos diferentes de zero de uma matriz e monta a estrutura especificada anteriormente. Considere que a entrada consiste dos valores de m e n (número de linhas e de colunas da matriz) seguidos de triplas ( i, j, valor) para os elementos diferentes de zero da matriz .
- TMatriz somaMatriz(TMatriz A, TMatriz B): Essa função recebe como parâmetros as matrizes A e B, devolvendo uma matriz C que é a soma de A com B.
- TMatriz multiplicaMatriz(TMatriz A, TMatriz B): Essa função recebe como parâmetros as matrizes A e B, devolvendo uma matriz C que é o produto de A por B.
- void insere(int i, int j, double v): Essa função insere um elemento com valor v na posição indicada por i e j. Caso o valor a ser inserido seja zero, a função remove o referido nó da matriz (pois o mesmo não precisa ser representado).
- Uma interface orientada a menus: que permita a leitura das matrizes e a realização das operações anteriores.

- Agora você deve realizar um upgrade neste aplicativo. Nesta nova versão do programa devem ser acrescentadas as seguintes características:
- Salvar: Cada matriz criada dentro do programa deverá ser salva em um arquivo em formato binário, com a extensão .smd (sparse matrix data). (5 pontos)
- Abrir: Deverá ser possível carregar os dados armazenados em arquivos .smd para dentro do programa, para permitir o seu processamento (5 pontos).
- Exportar: Deverá permitir a exportação da matriz em formato texto (.txt) das matrizes, escrevendo-as integralmente como um texto, com cada elemento da matriz formatado com 4 casas inteiras e 3 decimais, na forma de linhas e colunas, e considerando os elementos com valor igual a zero, conforme o exemplo abaixo (o símbolo □ representa espaço em branco deixado pela formatação) (5 pontos):
 

```

      □□□0.000 □□□0.000 □□□0.000 □□□0.001
      □□□0.000 □□□0.000 □□□0.002 □□□0.000
      □□□0.000 □□□0.003 □□□0.000 □□□0.000
      □□□0.004 □□□0.000 □□□0.000 □□□0.000
      
```
- Importar: Deverá ler um arquivo texto .txt e criar uma matriz esparsa a partir dos dados no arquivo texto (5 pontos).
- Obs: Os nomes dos arquivos usados em todas as operações devem ser definidos pelo usuário.

---

## Questão 2 (30 pontos): Arquivos Binários - Controle de Biblioteca

Você foi contratado por uma biblioteca para desenvolver um sistema simples de controle de empréstimos. O sistema deverá manter informações atualizadas a respeito dos usuários da biblioteca e dos livros que a mesma possui. O sistema deverá manipular três arquivos:

- Arquivo de Usuários: deverá conter as informações pessoais de cada usuário. As informações necessárias para cadastro na biblioteca são as seguintes:
  - Matrícula: um inteiro de até 4 dígitos. Servirá como chave de pesquisa. Deve ser único para cada usuário, ou seja, não pode haver dois usuários com números de matrícula iguais.
  - RG: um inteiro de até 8 dígitos. Também deve ser único para cada usuário.

- CPF: uma string numérica de 11 dígitos. Também deve ser único para cada usuário.
  - Nome: uma string com no máximo 50 caracteres. Deve armazenar o nome do usuário.
  - Endereço: uma string com no máximo 70 caracteres. Deve armazenar o endereço do usuário (rua e número).
  - Data de nascimento: um inteiro de 8 dígitos, no formato DDMMAAAA.
  - Sexo: um caractere que pode assumir apenas os valores "M" e "F".
  - Bairro: uma string com no máximo 30 caracteres.
  - Telefone residencial: um inteiro de até 8 dígitos
  - Telefone Celular: um inteiro de até 8 dígitos.
  - Data de adesão: um inteiro de 8 caracteres, no formato DDMMAAAA. Representa a data em que o usuário se cadastrou na biblioteca.
  - Status: um inteiro que indica a situação atual do usuário, podendo assumir os seguintes valores:
    - 0 - Normal;
    - 1 - Suspenso.
  - Término da suspensão: Um inteiro de 8 caracteres, no formato DDMMAAAA, que indica a data de término da suspensão, caso o usuário esteja suspenso.
  - Volumes Emprestados: Um inteiro no intervalo [0, 4], que indica a quantidade de livros emprestados pelo usuário.
- Arquivo de Obras: Deverá conter as informações referentes a cada livro que a biblioteca possui. Os dados referentes aos livros são os seguintes:
- Código: um inteiro de 5 dígitos que identifica cada volume de maneira única. Deve ser único para cada volume cadastrado.
  - ISBN: uma string alfanumérica de até 12 caracteres. O ISBN é o mesmo para todos os volumes de um mesmo título.
  - Título: uma string com até 80 caracteres.
  - Autores: uma string com até 100 caracteres.
  - Assunto: uma string com até 80 caracteres.
  - Editora: uma string com até 50 caracteres
  - Ano da publicação: um inteiro de 4 dígitos.
  - Edição: Um inteiro de 1 dígito.
  - Status: um inteiro que indica o estado atual do livro. Pode assumir os seguintes valores:
    - 0 - Disponível para empréstimo;
    - 1 - Emprestado;

- 2 - Em manutenção;
- Arquivo de empréstimos: Armazena todos os empréstimos efetuados, relacionando os usuários com os volumes emprestados. Possui os seguintes campos:
  - Matrícula: Identifica o usuário que fez o empréstimo.
  - Código: Identifica o livro emprestado.
  - Data de Empréstimo: um inteiro de 8 dígitos no formato DDMMAAAA, que indica a data em que o livro foi emprestado.
  - Data de Devolução: Indica a data em que o livro deve ser devolvido.

De posse destas informações, você deve escrever um programa que faça o controle dos usuários, livros e empréstimos, de acordo com a seguinte especificação:

- O menu inicial do sistema deve ter as seguintes opções:
  - Cadastro de Usuários 2 pontos
  - Cadastro de Obras 2 pontos
  - Empréstimos e Devoluções 2 pontos
  - Relatórios 2 pontos
  - Total 8 pontos
- O módulo de Cadastro de Usuário, por sua vez, deve realizar as seguintes operações:
  - Cadastrar novo usuário 2 pontos
  - Atualizar cadastro 2 pontos
  - Consultar usuário (por matrícula) 2 pontos
  - Apagar usuário (físicamente) 2 pontos
  - Total 8 pontos
- A opção de cadastrar novo usuário deve verificar as seguintes restrições:
  - Impedir a entrada de valores repetidos nos dados de usuário que devem ser únicos.
  - Um usuário só pode ser apagado somente se não tiver mais nenhum empréstimo vinculado a ele.
- O módulo de cadastro de obras deve realizar as seguintes operações:
  - Cadastrar nova obra 2 pontos
  - Alterar status 2 pontos
  - Consultar obra (por título) 2 pontos
  - Apagar obra (físicamente) 2 pontos
  - Total 8 pontos

- A opção de cadastrar nova obra deve verificar as seguintes restrições:
  - Impedir a entrada de valores repetidos nos dados que devem ter valor único.
- A Opção de alteração de status só poderá alterar o status de "disponível" para "em manutenção" e vice-versa. Livros emprestados somente poderão ter seu status modificado através do módulo de empréstimo.
- A Operação de apagar obras somente poderá ser realizada para livros com status "disponível" e "em manutenção". Nenhum livro poderá ser apagado enquanto houver empréstimos vinculados a ele.
- O módulo de Empréstimos e Devoluções deve realizar as seguintes operações:
  - Emprestar: 2 pontos
    - Deve-se solicitar a matrícula do usuário e verificar se o mesmo encontra-se ou não suspenso. Isto inclui checar o arquivo de empréstimos e atualizar a situação do usuário, conferindo a data atual com a data de cada livro emprestado por ele. Caso algum livro esteja em atraso, atualizar a situação do usuário para "suspenso". A data do fim da suspensão deve ser contada como 30 dias a partir do dia seguinte a data de devolução em atraso mais recente. Caso o mesmo se encontre suspenso, deve-se imprimir a mensagem: "Empréstimo não pode ser realizado. Usuário suspenso até DD/MM/AAAA."
    - Deve-se verificar também o número máximo de volumes que o usuário pode manter emprestados simultaneamente (apenas 4). Se este máximo for alcançado, imprimir a mensagem: "Empréstimo não pode ser realizado: Usuário já possui 4 volumes em seu poder."
    - Caso o usuário não esteja suspenso, deve-se solicitar o código do livro desejado, e verificar se o mesmo encontra-se disponível. Se o livro não estiver disponível, imprimir a mensagem: "Empréstimo não pode ser realizado. Volume indisponível no momento."
    - Caso o código do livro informado não exista, imprimir a mensagem: "Empréstimo não pode ser realizado. Volume inexistente."

- Caso o usuário não esteja suspenso e o livro estiver disponível, deve-se inserir um registro no arquivo de empréstimos, solicitando apenas a data de devolução. O campo "Volumes emprestados" do arquivo de usuários deve ser atualizado. A situação do livro deve ser modificada para "emprestado".
- Devolver: **2 pontos**
  - Solicitar o código do livro a ser devolvido.
  - Caso o código do livro não exista, imprimir a mensagem "Erro: Código inválido."
  - Consultar o arquivo de empréstimos e verificar qual o usuário que realizou o empréstimo. Atualizar o número de volumes emprestados do usuário.
  - Se a data de devolução efetiva for posterior a data de devolução no arquivo de empréstimos, atualizar a situação do usuário para "Suspenso". Atualizar a data de término da suspensão do registro do usuário para 30 dias após a data atual. consultar no arquivo de obras qual o livro que foi emprestado e atualizar o seu status para disponível.
- Total **4 pontos**

- O módulo de relatórios deve imprimir em um arquivo texto o seguinte relatório: **2 pontos**

```
Relatório de Livros Emprestados                                Data: dd/mm/aaaa
(em branco)
Título      | Código | Usuário      | Devolução  | Telefone
-----
(80 char) | 99999 | (Nome 50 char)| dd/mm/aaaa | 9999-9999
... ..
... ..
-----
```

- As 4 primeiras linhas representam o cabeçalho do relatório, que deve ser impresso em cada página. As linhas 5 – 49 representam a listagem dos dados, do relatório, que devem estar ordenados pelo campo título. A linha 50 representa a linha que fecha a página do relatório. Após a última linha deve-se imprimir uma quebra de página.
- Para imprimir uma quebra de página em C, utiliza-se o caractere de escape "\f".