



MC 102 - ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES
 TERCEIRA LISTA DE EXERCÍCIOS

1 Matrizes

1. Execute o comando (faça sem o computador):

```
for (i=0;i<3;i++)
  for (j=1;j<3;j++)
    mat1[i][j] = mat[mat[i][j]][mat[j][i]];
```

Supondo que todas as variáveis tenham sido declaradas corretamente, responda às perguntas:

- (a) Qual o valor da variável **mat1** resultante?
 - (b) Substitua no comando a variável **mat1** por **mat**. Qual o valor de **mat** resultante?
2. Escreva um programa que, dada uma matriz de m linhas e n colunas, verifique se esta é simétrica.
 3. Uma matriz de elementos inteiros é um *quadrado mágico* se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais. Exemplo: a matriz abaixo é um quadrado mágico.

$$A = \begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$$

Escreva um programa que verifique se uma matriz de n linhas e n colunas é um quadrado mágico.

4. No tabuleiro do jogo busca-minas existem dois tipos de células: as que contêm minas e as que não. As células que não têm minas armazenam o número de minas em sua vizinhança. Dada uma matriz de inteiros na qual as minas são representadas pelo valor 99, calcule os valores das demais células. Por exemplo:

$$\begin{pmatrix} 99 & 0 & 0 & 0 & 0 & 0 \\ 0 & 99 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 99 & 99 & 0 & 0 \\ 99 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 99 & 2 & 1 & 0 & 0 & 0 \\ 2 & 99 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 \\ 1 & 2 & 99 & 99 & 1 & 0 \\ 99 & 2 & 2 & 2 & 1 & 0 \end{pmatrix}$$

5. Dado um elemento $mat[i][j]$ de uma matriz mat , dizemos que os elementos *adjacentes* a $mat[i][j]$ são $mat[i-1][j-1]$, $mat[i-1][j]$, $mat[i-1][j+1]$, $mat[i][j-1]$, $mat[i][j+1]$, $mat[i+1][j-1]$, $mat[i+1][j]$ e $mat[i+1][j+1]$. Note que $mat[1][1]$ tem somente três elementos adjacentes, o mesmo acontecendo com $mat[1][n]$, $mat[n][1]$ e $mat[n][n]$. Analogamente, os outros elementos da primeira e última linha e coluna de mat tem somente 5 elementos adjacentes. Todos os outros elementos de mat tem 8 elementos adjacentes. Escreva um programa que leia uma matriz mat de números inteiros, de m linhas e n colunas e produza uma matriz $mat1$, também de m linhas e n colunas, tal que $mat1[i][j]$ contenha a média dos elementos adjacentes a $mat[i][j]$.
6. A matriz abaixo representa o Triângulo de Pascal de ordem 6:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```

Os elementos extremos de cada linha são iguais a 1. Os outros elementos são obtidos somando-se os dois elementos que aparecem imediatamente acima e à esquerda na linha anterior. Assim, $10 = 4 + 6$.

Escreva três versões de um programa que, dado n , construa e imprima o Triângulo de Pascal de ordem n , utilizando:

- (a) Uma matriz
 - (b) Dois vetores
 - (c) Apenas um vetor
7. Uma matriz de permutações é uma matriz quadrada cujos elementos são zeros ou uns tal que em cada linha e em cada coluna exista *exatamente* um elemento igual a 1. A matriz abaixo representa uma matriz de permutações de ordem 3:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Escreva um programa que determine se uma matriz lida do teclado é uma matriz de permutações.

8. Suponha que tenham sido feitas as seguintes declarações:

```
#define MAX 20
```

```
typedef int[MAX][MAX] Tpmatriz
```

Escreva uma função que receba uma matriz como parâmetro e retorne, simultaneamente: (i) o maior elemento da matriz; (ii) o menor elemento da matriz (iii) o elemento que mais se aproxima de seu valor médio.

9. Uma matriz esparsa é um vetor bidimensional cuja maioria dos elementos é zero. É muito caro armazenar tal matriz, pois poucos elementos possuem informações significativas. Um modo mais eficiente de representar uma matriz esparsa $m \times n$ é armazenar apenas informações sobre os elementos não nulos. Para isso, usamos uma matriz, de tamanho $k \times 3$, na qual armazenamos o valor, os índices da linha e da coluna somente dos elementos não nulos. Esta é a chamada *representação reduzida* da matriz esparsa.

Por exemplo, a matriz

$$\begin{pmatrix} 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & -8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

pode ser representada como:

$$\begin{pmatrix} 1 & 3 & 7 \\ 2 & 4 & -8 \\ 4 & 1 & 2 \end{pmatrix}$$

Escreva uma função **void** RepReduzida (**int** m, **int** n, **Tpmatriz** matred) que leia uma matriz esparsa $m \times n$ e construa sua representação reduzida, devolvendo o resultado no parâmetro *matred*. Escreva as declarações adequadas para *Tpmatriz*. Como não será devolvido o tamanho da matriz *matred* gerada, preencha sua última linha com zeros.

10. Preencha os retângulos abaixo de tal modo que após a execução do trecho do programa, cada elemento de *c* contenha o mínimo dos elementos da respectiva linha de *a* e cada elemento de *b* contenha o mínimo dos elementos da respectiva coluna de *a*. Suponha que a constante *MAX* tenha sido definida e que as variáveis *m*, *n* e *a* tenham sido iniciadas.

```

int a[MAX][MAX], b[MAX], c[MAX];
int i, j, n, m;
...
for (i=0; i<m; i++) ;
for (i=0; i<n; i++) c[i]=a[i][0];

for (i=0; i<n; i++)
  for (j=0; j<m; j++) {
    if (  ) ;
    if (  ) ;
  }
...

```

2 Registros

1. Para cada um dos seguintes itens, projete uma estrutura de registro apropriada para as informações dadas¹.

- As cartas em um maço de jogo de baralho.
- Tempo medido em horas, minutos e segundos.
- Os registros de uma lista telefônica.
- Descrição de um automóvel (Marca, Ano, Modelo, Tipo, Cor, Acessórios, Preço).
- Descrição de um livro na biblioteca (Autor, Título, Editora, etc.).
- Times de futebol no campeonato nacional (Time, grupo, número de partidas jogadas, etc.).
- Posição das pedras em um jogo de damas no tabuleiro.

2. Para cada um dos exemplos a seguir, crie o seu registro equivalente ².

Pessoas

DADOS PESSOAIS

Nome: Miguel João da Cruz

Telefone: 714.3914

Aniversário: 01 / 01 / 1951

Cidade: Brasília

Estado: DF

Filmes

FILME EM CARTAZ

Título: Amores Eletrônicos

Duração: 120'

Seções: 10:00 - 13:30 -

15:40 - 18:00 - 21:30

Médicos de plantão

Matrícula	Nome	Hor. Inicial	Hor. Final	Especialidade
18.001-9	Rilmar Gomes	18:00	0:00	Clínico Geral
27.653-1	Fernando Soares	0:00	6:00	Dermatologia
19.753-8	Manoel Soares	6:00	18:00	Cardiologia

¹Fonte: http://www-usr.inf.ufsm.br/~pozzer/disciplinas/ed_2_exerc_revisao.pdf

²Fonte: http://www.unifra.br/professores/14283/ED_02_Registro_ex.pdf

3. A equação de uma reta com coeficiente angular m e passando pelo ponto P com coordenadas (x_1, x_2) é

$$y - y_1 = m(x - x_1)$$

- (a) Escreva uma descrição de um registro para uma reta, dados sua inclinação (coeficiente angular) e um ponto na reta.
 - (b) Escreva um programa que leia a inclinação da reta e as coordenadas de um ponto na reta e então encontre a equação da reta.
 - (c) Escreva um programa que leia o ponto e a inclinação para duas retas e determine se elas se interceptam ou são paralelas. Se elas se interceptam encontrar o ponto de interseção e determinar se elas são perpendiculares.
4. Os números flutuantes armazenam valores com casas decimais, mas não são apropriados para certas situações devido a sua margem de erro. Por exemplo, em aplicativos financeiros, qualquer margem de erro é igual a dinheiro perdido. Nesse caso, o que você precisa é armazenar a parte inteira em um `int` e a parte decimal em outro `int`. Escreva um programa que armazene os números daquela maneira e implemente as seguintes operações aritméticas: adição, subtração e multiplicação.
5. Escreva um programa que permita cadastrar imóveis (no máximo 30) a serem alugados ou vendidos, contendo os seguintes dados: tipo (loja, apartamento, casa, kit), endereço, bairro, valor, situação (aluguel ou venda). Ao final, solicitar ao usuário a situação a ser pesquisada (tipo, bairro, valor ou situação) e mostrar todos os dados dos imóveis enquadrados na solicitação ³.

3 Arquivos

1. Escreva um programa que, dado o nome de um arquivo de texto, imprima na tela o número de caracteres, de linhas e o de palavras que contem.
2. Escreva um programa que receba do usuário o nome de um arquivo de texto e crie uma cópia deste com o mesmo nome mas o sufixo “back”. Por exemplo, se o arquivo de entrada é “entrada.txt”, a cópia tem se que chamar “entrada.txt.back”.
3. Escreva um programa que leia um arquivo de números float, um número em cada linha, e mostre na tela qual é o número maior e o numero menor e em que linha foram encontrados. Por exemplo, no arquivo:

```
12.5
33.0
0.2
-236.1
33.0
```

³Fonte: http://www.unifra.br/professores/14283/ED_02_Registro_ex.pdf

24.519
14.0
-236.05

a saída esperada é:

Número maior: 33.0 na linha 2.
Número maior: 33.0 na linha 5.
Número menor: -236.1 na linha 4.

No máximo o arquivo vai conter 10 cópias do mesmo número.

4. Considere arquivos com conteúdo similar ao seguinte:

```
4
GABRIEL SA
7.5
6.0
10.0
DAVI SANTOS SANTOS
7.0
9.1
8.5
MIGUEL OLIVEIRA
5.0
5.5
6.0
ARTHUR SILVA
```

A primeira linha contém o número de alunos. As linhas seguintes contém o nome do aluno (no máximo 80 caracteres) e as notas de suas 3 provas. Escreva um programa que imprima o nome e a média dos alunos com média acima de 7⁴.

5. Em uma eleição presidencial existem quatro candidatos. Os votos são informados através dos seguintes códigos:

1, 2, 3, 4 ⇒ votos para os respectivos candidatos;
5 ⇒ voto nulo;
6 ⇒ voto em branco.

Suponha que os votos tenham sido armazenados em um arquivo. Faça um programa que simule uma eleição (carregando o arquivo de votos) e posteriormente calcule as seguintes informações:

- (a) Total de votos para cada candidato, de votos brancos e nulos.

⁴Fonte: <http://equipe.nce.ufrj.br/adriano/c/apostila/es.htm#exercicios>

- (b) Porcentagem de votos para cada candidato sobre o total de votos válidos (sem considerar brancos e nulos).
- (c) Porcentagem de votos nulos sobre o total de votos;
- (d) Porcentagem de votos brancos sobre o total de votos.

Seu programa deve ter as seguintes funções:

float totaliza(**int** votos[], **int** código) e
float porcentagem(**int** total1, **int** total2).

A função *totaliza* recebe o vetor de votos e um código e retorna o número de votos com aquele código. A função *porcentagem* recebe o número de ocorrências de um conjunto de dados e o total sobre o qual a porcentagem deve ser calculada. Os dados solicitados devem ser calculados usando-se as funções *totaliza* e *porcentagem*.

6. Uma maneira de ocultar uma mensagem é enviar, além do texto que se quer transmitir, conteúdo extra sem valor. Porém, é necessário que o destinatário da mensagem possa reconhecer o conteúdo valioso. Suponha que você recebeu o arquivo seguinte:

ABOZXI4PBJMEMSFSORLQSAPL

A mensagem está oculta, mas você possui o com a sequencia de caracteres que têm que ser ignorados, por exemplo:

ABZX4BJMMFRLQSPIMKFXWYGHTDANRMKVORYXCAQMDFUWIEOARWFO

Para decodificar a mensagem você precisa ler o conteúdo de ambos arquivos simultaneamente é seguir as seguintes regras:

- (a) se o caracter atual em ambos arquivos é o mesmo, avança um caracter em ambos arquivos.
- (b) se o caracter atual do arquivo com a mensagem é distinto ao caracter atual do arquivo com o código, imprime na tela o caracter atual da mensagem e avança ao seguinte caracter naquele arquivo. O caracter atual do arquivo com o código não muda.
- (c) se não tiver mais caracteres pra ler no arquivo com a mensagem o programa termina.

Inicialmente o caracter “atual” de ambos arquivos é o primeiro caracter. No exemplo de acima, o resultado da execução é:

OIPESSOAL

Escreva um programa que reciba o nome do arquivo da mensagem e o nome do arquivo do código e imprima a mensagem oculta. Tenha em conta que os arquivos podem não ter a mesma quantidade de caracteres (no exemplo o arquivo com o código é maior ao arquivo com a mensagem, mas pode acontecer ao contrario).

7. “Head” é uma ferramenta que permite mostrar na tela só as *n* últimas linhas de um arquivo de texto. Escreva um programa que leia da tela o nome de um arquivo de texto e o valor de *n* e mostre as últimas *n* linhas dele.

8. “Tail” é uma ferramenta que permite mostrar na tela só as n últimas linhas de um arquivo de texto. Escreva um programa que leia da tela o nome de um arquivo de texto e o valor de n e mostre as últimas n linhas dele.
9. “Awk” é uma ferramenta que permite dividir o conteúdo de um arquivo em colunas e mostrar só algumas delas. Por exemplo, no arquivo de texto seguinte que armazena um registo das atividades no sistema:

```
10:20:32 - ROOT - Login attempt - 192.168.1.2 - FAILED
10:23:21 - GUESS - Start session - 192.168.1.2 - SUCCESS
10:23:25 - GUESS - Logout - 192.168.1.2 - SUCCESS
```

se você quiser mostrar só a hora, é possível escolher o caracteres “-” como o divisor das colunas, e logo faz que “Awk” só mostre a primeira coluna:

```
10:20:32
10:23:21
10:23:25
```

Agora, se você quiser mostrar só o nome do usuário que executó a ação, escolheria o mesmo caracteres “-”, mas mostraria a segunda coluna.

```
ROOT
GUESS
GUESS
```

Escreva um programa que leia da tela o nome de um arquivo de texto, um caracter c é um número n , e mostre na tela a n -ésima coluna do arquivo utilizando o caracter c como divisor das colunas.

10. “Diff” é uma ferramenta que permite comparar dois arquivos de texto. Por exemplo, seja o arquivo A:

```
Supõe tu um campo de batatas e duas tribos famintas. As batatas apenas
chegam para alimentar uma das tribos, que assim adquire forças para
transpor a montana e ir à outra vertente,
```

Seja o arquivo B:

```
Supõe tu campo de batatas e duas tribos faminta. As batatas la apenas
chegam para alimentar uma das tribos, que assim adquire forças para
transpor a montanha e ir à outra vertente,
onde há batatas em abundância;
```

A saída esperada é:

```

Arquivo A >
Arquivo B <
-- linha 1
> Supõe tu um campo de batatas e duas tribos famintas. As batatas apenas
< Supõe tu campo de batatas e duas tribos faminta. As batatas la apenas
-- linha 3
> transpor a montanha e ir à outra vertente,
< transpor a montana e ir à outra vertente,
-- linha 4
< onde há batatas em abundância;

```

Escreva um programa que leia da tela os nomes de dois arquivos de texto e compare seu conteúdo, mostrando na tela as diferenças entre eles. Se ambos arquivos forem idênticos, o programa não produz saída.

11. “Grep” é uma ferramenta que permite procurar padrões de texto em um ou vários arquivos. Escreva um programa que leia da tela o nome de um arquivo e a palavra a procurar. Por cada palavra encontrada a saída deve mostrar o número de linha e o seu conteúdo. Por exemplo, suponha que a palavra procurada é “tribos” e o texto do arquivo de entrada for:

```

Supõe tu um campo de batatas e duas tribos famintas. As batatas apenas
chegam para alimentar uma das tribos, que assim adquire forças para
transpor a montanha e ir à outra vertente,
onde há batatas em abundância;
mas, se as duas tribos dividirem em paz as batatas do campo,
não chegam a nutrir-se suficientemente e morrem de inanição

```

A saída deveria ser:

```

1: Supõe tu um campo de batatas e duas tribos famintas. As batatas apenas
2: chegam para alimentar uma das tribos, que assim adquire forças para
4: mas, se as duas tribos dividirem em paz as batatas do campo,

```

12. A “Cifra de César”, *“é uma das mais simples e conhecidas técnicas de criptografia. . . na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes”* (Wikipedia). Por exemplo, se o número for 5, a letra “**A**” seria trocada pela letra “**F**”, a “**B**” pela “**G**”, a “**V**” pela “**A**”, a “**W**” pela “**B**”, e assim por diante.
- Escreva um programa que dado o nome de um arquivo e um número N , gere outro arquivo com o mesmo conteúdo que o primeiro mas codificado.
 - Escreva um programa que dado o nome de um arquivo codificado e um número N , gere outro arquivo com o mesmo conteúdo que o primeiro mas decodificado.

Arquivos binários

13. Faça um programa que simule um controle de estoque de uma loja, onde cada produto, representado por meio de um registro, possui um identificador inteiro, nome, quantidade e custo unitário. O programa deve permitir a inclusão e remoção de novos produtos, consulta de produtos por nome, alteração de registros, geração de relatórios (ex: qual vendeu mais, qual tem maior estoque, produtos cujo estoque estejam abaixo de X unidades, etc.). Todos os registros devem ser armazenados sequencialmente em um arquivo binário. O programa deve disponibilizar um menu de opções ao usuário. Ex: 1 - Incluir Produto, 2 - consultar, etc⁵.
14. Elabore um programa que leia dois arquivos binários (salbruto.bin e desc.bin) contendo, respectivamente, os salários brutos e os descontos de até 100 funcionários. Para cada registro, deve-se calcular o salário líquido (salário bruto - descontos) e gravá-lo no arquivo texto com nome salario-liq.txt⁶.
15. Dada o seguinte registro

```
struct employee {
    char last_name[20];
    char first_name[15];
    int age;
    float salary;
};
```

- (a) Escreva um programa que reciba os dados dos trabalhadores e os armazene sequencialmente em um arquivo binário.
- (b) Escreva um programa que leia o arquivo gerado pelo programa anterior e permita fazer a consulta dos dados de um trabalhador. O programa terá duas opções:

1- Mostrar dados do um trabalhador.
2- Sair

A opção (1) recebe como entrada o índice do trabalhador. Por exemplo, o índice do primeiro trabalhador inserido no programa anterior é 1, do segundo é 2, e assim por diante. A saída do seu programa deve ter o seguinte formato:

```
TRABALHADOR #6
-----
Nome: XXXXXX
Sobrenome: XXXX
Idade: XXXXXX
Sueldo: R$ XX,XX
```

Uma restrição do programa é que não pode armazenar na memória mais de um registro por vez, portanto, você não pode criar um vetor e armazenar todos os registros no começo⁷.

⁵Fonte: http://www-usr.inf.ufsm.br/~pozzer/disciplinas/ed_2_exerc_revisao.pdf

⁶Fonte: http://www.facom.ufu.br/~gustavo/IC/Programacao/2a_Lista_Exercicios.pdf

⁷Fonte: www.inf.udec.cl/~leo/fio.pdf