

# Linguagem de Programação 2

## Lista de Exercícios 5 - Matrizes e Vetores

Prof. Flávio José Mendes Coelho<sup>1</sup>  
fcoelho@uea.edu.br  
est.uea.edu.br/fcoelho

1. Seja a matriz

$$\mathbf{M} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix}.$$

Escreva um programa que declare a matriz **M** de forma genérica utilizando as constantes LINHA e COLUNA para definir suas dimensões (sua ordem), e imprima os elementos de **M** de acordo com as seguintes sequências de impressão:

- a) a, b, c, d, e, f, g, h, i, j, k, l.
- b) l, k, j, i, h, g, f, e, d, c, b, a.
- c) a, e, i, b, f, j, c, g, k, d, h, l.
- d) l, h, d, k, g, c, j, f, b, i, e, a.
- e) d, c, b, a, h, g, f, e, l, k, j, i.
- f) i, j, k, l, e, f, g, h, a, b, c, d.
- g) d, h, l, c, g, k, b, f, j, a, e, i.
- h) i, e, a, j, f, b, k, g, c, l, h, d.
- i) a, b, c, d, h, g, f, e, i, j, k, l.
- j) l, k, j, i, e, f, g, h, d, c, b, a.

Faça um programa para cada sequência. Seu programa deverá funcionar para qualquer outra matriz de caracteres com dimensões quaisquer (suponha que a menor matriz possível tenha ordem 2x2).

2. Escreva um programa que inicialize duas matrizes **A** e **B** numéricas com valores quaisquer. O programa deve imprimir a matriz soma de **A** e **B**.

3. Escreva um programa que inicialize uma matriz de inteiros **A** de dimensão  $n \times n$  com valores quaisquer, e mostre todos os elementos da diagonal principal de **A**. Na diagonal principal estão

---

<sup>1</sup>Em colaboração com os professores do curso de Engenharia da Computação da EST/UEA.

os elementos  $a_{00}, a_{11}, \dots, a_{nn}$ .

4. Escreva uma nova versão do programa do problema anterior para apresentar a soma dos elementos da diagonal principal.
5. Escreva um programa que inicialize uma matriz de inteiros  $\mathbf{A}$  de dimensão  $n \times n$  com valores quaisquer, e mostre todos os elementos da diagonal secundária de  $\mathbf{A}$ . Na diagonal secundária fica oposta à diagonal principal.
6. Escreva um programa que inicialize uma matriz  $\mathbf{A}$  com valores quaisquer, e leia um valor  $x$ . O programa deve contar quantas vezes  $x$  ocorre em  $\mathbf{A}$ , e imprimir este valor ao final do programa.
7. Escreva um programa que inicialize uma matriz  $\mathbf{A}$  com valores quaisquer, e uma matriz  $\mathbf{B}$  com valores nulos. O programa deve copiar da matriz  $\mathbf{A}$  para a matriz  $\mathbf{B}$ , todos os valores negativos de  $\mathbf{A}$ , para suas posições correspondentes na matriz  $\mathbf{B}$ .
8. Escreva um programa que inicialize uma matriz  $\mathbf{A}_{m \times n}$  com valores quaisquer, e inicialize um vetor  $v$  de tamanho  $mn$  com valores nulos. O programa deve copiar todos os valores de  $\mathbf{A}$  para  $v$ .
9. Escreva uma nova versão do programa do problema anterior para copiar todos os valores do vetor  $v$  para a matriz  $\mathbf{A}$ .
10. Escreva um programa que inicialize uma matriz  $\mathbf{A}_{m \times n}$  com valores quaisquer, e que declare uma matriz  $\mathbf{B}_{m \times n}$ , ambas com mesma ordem e mesmo tipo. O programa deve copiar todos os valores de  $\mathbf{A}$  para  $\mathbf{B}$ , na ordem inversa, ou seja,  $a_{11} = b_{nn}, a_{12} = b_{m-1n}$ , etc, até que  $a_{mn} = b_{11}$ . Mostre a matriz  $\mathbf{B}$  após a cópia.
11. Escreva um programa que inicialize uma matriz com valores quaisquer, e leia um valor  $x$ . O programa deve realizar uma busca na matriz à procura da primeira ocorrência de  $x$ , e imprimir a posição  $(i, j)$  onde o  $x$  foi encontrado. E se  $x$  não fosse encontrado? O que modificação você faria no programa?
12. Escreva um programa que inicialize uma matriz  $\mathbf{A}_{m \times n}$ . O programa deve calcular a média de todos os elementos de  $\mathbf{A}$ , e imprimir a média, ao final do programa.
13. Escreva um programa que inicialize duas matrizes  $\mathbf{A}_{m \times n}$  e  $\mathbf{B}_{n \times q}$ . O programa deve gerar a matriz produto  $\mathbf{A} \cdot \mathbf{B}_{m \times q}$ , e mostrá-la ao final do programa. Lembre-se:  $(AB)_{ij} = \sum_{k=1}^n a_{ik}b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$ .
14. Escreva um programa que inicialize uma matriz, e que leia um valor  $k$ . O programa deve multiplicar todos os elementos da matriz por  $k$ , e mostrá-la ao final.
15. Escreva um programa que inicialize uma matriz  $\mathbf{A}_{m \times n}$  e que gere uma outra matriz  $\mathbf{T}_{n \times m}$  que seja a matriz transposta de  $A$ . A matriz transposta de uma matriz  $A$  possui como linhas as colunas correspondentes da matriz  $A$ . Imprima as matrizes  $A$  e  $T$ , ao final do programa.
16. Escreva um programa que inicialize uma matriz  $\mathbf{A}$ , e que leia um valor  $k$ . O programa deve

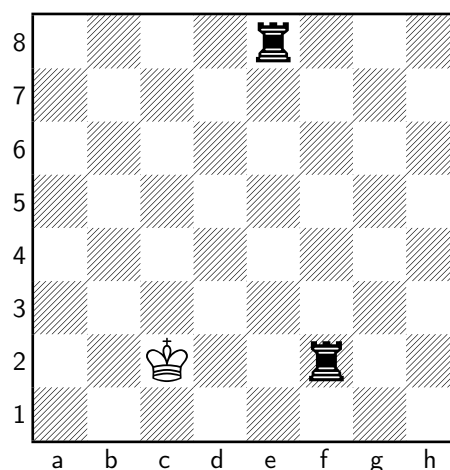
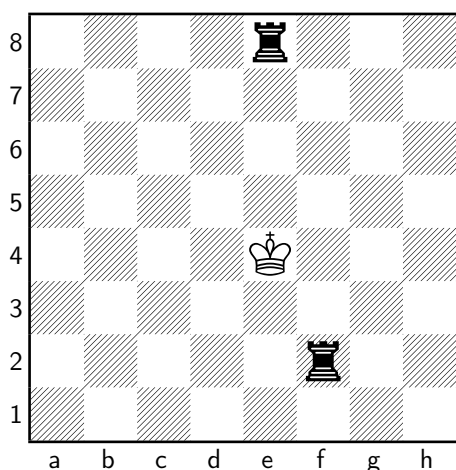
zerar todos as posições onde o valor  $k$  ocorrer na matriz. Mostre como a matriz ficará ao final do programa.

17. Escreva um programa que inicialize uma matriz  $\mathbf{A}$ , e imprima todos os elementos que estão acima da diagonal principal.

18. Considere uma matriz  $\mathbf{X}_{8 \times 8}$  que representa um tabuleiro do jogo de xadrez. Cada posição  $(i, j)$  da matriz armazena somente um de três valores possíveis:

- ' ' (caracter com espaço em branco), indicando que nesta posição não há peça alguma;
- 'R' (caracter 'R'), indicando que nesta posição está o rei branco;
- 't' (caracter 't'), indicando que nesta posição há uma torre negra;

Considere que o tabuleiro já foi inicializado e que há nele, um rei branco e duas torres negras em posições distintas (atribua manualmente estas posições). Considere que a linha e a coluna onde o rei se encontra no tabuleiro estarão armazenadas, respectivamente, nas variáveis `linRei` e `colRei`, assim como as posições das torres também estarão armazenadas nas variáveis `linT1` e `colT1` para a primeira torre, e `linT2` e `colT2` para a segunda torre. Escreva um programa que verifique se o rei está sob ameaça de alguma torre. Por exemplo, se o rei na posição (3, 5) estiver sendo ameaçado por uma torre na posição (3, 4), o programa deverá mostrar na tela a mensagem "Rei (3, 5) ameaçado pela torre (3, 4)". O rei está sob ameaça quando há uma torre na linha e/ou na coluna que intercepta o rei. Na primeira figura abaixo, o rei está sob ameaça pois está na mesma coluna da torre na posição e8. Na segunda figura, a ameaça ocorre porque o rei está na mesma linha da torre em f2.



Se o rei não estiver sob ameaça, mostre uma mensagem para indicar esta situação. Desenvolva o programa de tal forma que se as peças forem mudadas de posição, o programa continuará funcionando sem a necessidade de outras modificações.

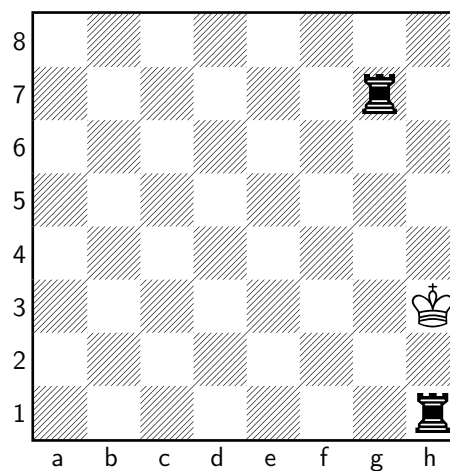
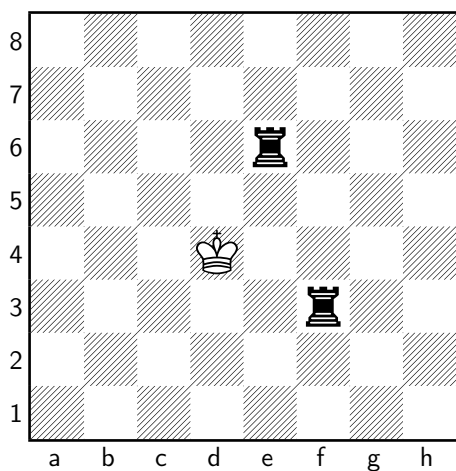
19. Crie uma nova versão do programa do problema anterior para mostrar o tabuleiro na tela no formato de tabela, como mostra a figura abaixo.

```

      1 2 3 4 5 6 7 8
    +-+-+-+-+-+-+-+
1|  |  |  |  |  |t|  |  |
    +-+-+-+-+-+-+-+
2|  |  |  |  |  |  |  |
    +-+-+-+-+-+-+-+
3|  |  |  |  |R|  |  |
    +-+-+-+-+-+-+-+
4|  |  |t|  |  |  |  |
    +-+-+-+-+-+-+-+
5|  |  |  |  |  |  |  |
    +-+-+-+-+-+-+-+
6|  |  |  |  |  |  |  |
    +-+-+-+-+-+-+-+
7|  |  |  |  |  |  |  |
    +-+-+-+-+-+-+-+
8|  |  |  |  |  |  |  |
    +-+-+-+-+-+-+-+
    
```

20. Modifique o problema do xadrez para que as posições do rei e das torres sejam descobertas pelo programa, antes da verificação de ameaça do rei.

21. Modifique o problema do xadrez para que o rei sob ameaça faça um movimento de fuga para uma posição segura, isto é, para uma casa do tabuleiro adjacente à sua posição atual, que não esteja sob ameaça das torres. Neste caso, mostre o tabuleiro antes e depois do movimento. Se não for possível encontrar um posição segura, mostre o tabuleiro e emita a mensagem “Xeque-mate!”. Abaixo, a figura à esquerda ilustra um caso em que o rei está seguro, e a figura à direita um caso em que o rei está sob xeque-mate.



22. An rede internacional de *fast-food* JaraquiDonalds prepara-se cinco tipos de sanduíches usando cinco ingredientes em proporções variadas, conforme a tabela, a seguir:

	Ova de Piranha	Queijo de Pacú	Presunto de Tucunaré	Hamburguer de Jaraqui	Vinagrete
<b>Jara-burger</b>	0	1 fatia	0	1	0
<b>Jara-misto</b>	0	2 fatias	1 fatia	0	0
<b>Jara-salada</b>	70g	1 fatia	2 fatias	1	1 porção
<b>Jara-tudo</b>	100g	2 fatias	1 fatia	1	2 porções

Os preços unitários dos ingredientes constam da tabela seguinte:

Ingrediente	Preço (R\$)
Ova de Piranha	12,0/Kg
Queijo de Pacú	10,0/Kg
Presunto de Tucunaré	8,5/Kg
Hamburguer de Jaraqui	13,5/Kg
Vinagrete	3,0/porção

Considere que uma fatia de queijo ou presunto possui 15g, escreva um programa que calcule o preço total de cada sanduíche, mostrando estes preços ao final do programa.

23. (Baseada na Prova EST/2010-2) Um **quadrado mágico** é uma matriz quadrada em que a soma de qualquer uma de suas linhas, colunas, diagonal principal e secundária resulta em um mesmo valor. A matriz abaixo é um exemplo de quadrado mágico, pois as somas de linhas, colunas, etc, em todos os casos, é igual a 15.

4	9	2
3	5	7
8	1	6

Escreva um programa que leia uma matriz de tamanho  $3 \times 3$  de inteiros distintos (de 1 a 9) e, ao final, imprima se esta matriz é ou não um quadrado mágico.

24. Generalize o programa do quadrado mágico para verificar se uma matriz quadrada de inteiros de ordem maior que 1 é um quadrado mágico.

25. (Baseado na Prova EST/2010-2) Escreva um programa em que o usuário preencha uma matriz de tamanho  $10 \times 5$  com valores inteiros. Em seguida, o programa deverá selecionar os valores da matriz que sejam múltiplos de 7 e armazenar esses valores em um vetor. Ao final, imprima o vetor com os valores múltiplos de 7 que estavam na matriz.

26. (Baseado na Prova EST/2011-1) Faça um programa para ler uma matriz  $100 \times 100$  de inteiros positivos. Em seguida, verifique, somente para as linhas pares da matriz, quais os elementos são múltiplos de 5 e coloque-os em um vetor. Ao final imprima o vetor resultante somente com as posições preenchidas. Observações: (1) o vetor deve ser preenchido sequencialmente; (2) não faça verificação dos múltiplos de 5 junto com a leitura da matriz.

27. (Baseado na Prova EST/2011-1) Faça um programa para ler uma matriz  $100 \times 100$  de inteiros. Em seguida, nas linhas ímpares, o programa deverá capturar os valores pares encontrados e colocá-los, se existirem, em um vetor **pares**, trocando estes valores na matriz pelo número 1. Em seguida, para as linhas pares da matriz, o programa deverá copiar os valores ímpares encontrados, se existirem, para um vetor **ímpares** e trocar, na matriz, estes valores pelo número 2. Ao final o programa deve imprimir a

matriz e os vetores (somente com as posições preenchidas dos vetores).

28. Escreva um programa que leia dois vetores de caracteres de tamanhos iguais, e verifique se estes são palíndromos. Palíndromo é uma palavra, frase ou qualquer sequência de caracteres que tenha a propriedade de ser lida da direita para esquerda ou da esquerda para a direita e o conteúdo representar a mesma palavra. Exemplos de palíndromos: *arara*, *radar* e *mirim*.

29. Escreva um programa que leia duas palavras de tamanhos iguais e verifique se estas são anagramas. Um anagrama é arranjo das letras de uma palavra (ou frase) para produzir outras palavras, utilizando todas as letras originais exatamente uma vez. Diferentemente dos palíndromos, o arranjo não precisa ser igual, pois todo palíndromo é um anagrama, mas nem todo anagrama é um palíndromo. Exemplos de anagrama: *Célia* e *Alice*, e *Lyseu* e *Suely*.

30. Faça um programa que leia um vetor  $A$  de inteiros, e copie para um vetor  $B$  de inteiros, todos os itens do vetor  $A$ , sem repetições de itens. Mostre ambos os vetores, ao final.

31. Um algoritmo de compactação toma um arquivo como entrada e cria uma cópia reduzida (mais compacta) do arquivo original economizando espaço de memória. Exemplos importantes de arquivos compactados são os arquivos de áudio, **mp3**, e os de vídeo, **mpeg**. Escreva um programa para compactar um texto (*string*) armazenada em um vetor de entrada, gerando um vetor de saída com uma versão compactada (com menos caracteres) do texto original. Utilize o seguinte algoritmo de compactação:

- 1) Tome o primeiro caracter  $\alpha$  do vetor de entrada.
- 2) Verifique cada posição onde  $\alpha$  ocorre no vetor de entrada, e armazene no vetor de saída (de inteiros) a seguinte sequência de valores: o número 0 (zero), seguido do valor do caracter  $\alpha$  convertido em inteiro, seguido das posições de ocorrência de  $\alpha$ . O número zero serve apenas como um marcador para separar um caracter sendo compactado, do próximo a ser compactado, e assim, por diante.
- 3) Armazene  $\alpha$  em um vetor temporário de caracteres (mesmo tamanho do vetor de entrada).
- 4) Pegue o próximo caracter  $\alpha$  no vetor de entrada. Verifique se este caracter existe no vetor temporário. Se sim, descarte-o, e continue no passo 4 pegando o próximo caracter da entrada. Se  $\alpha$  não existir no vetor temporário, então prossiga para o passo 5.
- 5) Retorne para o passo 2 até ter visitado todos os caracteres do vetor de entrada.

Ao final do programa, mostre o vetor de saída. Observe que o texto de saída, compactado, não é totalmente legível, pois conterá somente números separados por zeros. É necessário descompactar este texto gerando novamente o texto original.

32. Escreva um programa para descompactar um texto compactado pelo programa do problema anterior. O programa deve tomar o texto compactado como entrada e gerar o texto original como saída. Utilize uma lógica inversa à do algoritmo descrito no problema anterior.

33. Faça um programa que leia um vetor  $A$  de inteiros, e um inteiro  $k$  pertencente ao vetor. O programa deve particionar  $A$  em três porções: a primeira contém todos os elementos menores ou iguais a  $k$ ; a segunda possui  $k$ ; e a terceira, todos os elementos maiores que  $k$ . Não utilize outro vetor como auxiliar, e mostre o vetor  $A$ , ao final. Exemplo: Entrada:  $A = [4, 7, 2, 3, 6, 1, 5]$  e  $k = 5$ . Saída:  $A = [4, 2, 3, 5, 7, 6]$ .

34. Crie um programa que inicialize uma matriz de inteiros de ordem  $m \times n$ , e ordene-a em ordem não-decrescente, de forma que o elemento mínimo fique na posição  $(1, 1)$  e o elemento máximo fique na posição  $(m, n)$  da matriz.

35. Faça um programa que gere a seguinte matriz:

1	3	3	3	3	2
3	1	3	3	2	3
3	3	1	2	3	3
3	3	2	1	3	3
3	2	3	3	1	3
2	3	3	3	3	1

36. Faça um programa que gere a seguinte matriz:

1	1	1	1	1	1
1	2	2	2	2	1
1	2	3	3	2	1
1	2	3	3	2	1
1	2	2	2	2	1
1	1	1	1	1	1

Sonhei que resolvi todos  
os problemas dessa lista...  
Então, acordei e fui ver TV  
pra descansar.

