

Linguagem de Programação 2

Lista de Exercícios 8

Prof. Flávio José Mendes Coelho

FUNÇÕES, VETORES, MATRIZES E STRINGS

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** 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.

Codifique uma função para cada sequência a ser gerada. A função receberá a matriz como parâmetro. Note que seu programa deverá funcionar para qualquer matriz de caracteres com dimensões quaisquer (suponha que a menor matriz possível tenha ordem 2x2).

2. Escreva uma função que receba duas matrizes **A** e **B** numéricas como parâmetros e imprima a matriz soma de **A** e **B**. Crie uma outra função somente para imprimir a matriz soma em formato matricial.

3. Escreva uma função que receba uma matriz de inteiros \mathbf{A} de dimensão $n \times n$, e que mostre todos os elementos da diagonal principal de \mathbf{A} . Na diagonal principal estão os elementos $a_{00}, a_{11}, \dots, a_{nn}$.
4. Escreva uma nova versão da função do exercício anterior para retornar a soma dos elementos da diagonal principal.
5. Escreva uma função que receba uma matriz de inteiros \mathbf{A} de dimensão $n \times n$, e mostre todos os elementos da diagonal secundária de \mathbf{A} . Na diagonal secundária fica oposta à diagonal principal.
6. Escreva uma função que receba uma matriz \mathbf{A} e um valor x , e que retorne quantas vezes x ocorre em \mathbf{A} .
7. Escreva uma função que receba duas matrizes \mathbf{A} e \mathbf{B} como parâmetros, e que copie os valores de \mathbf{B} para \mathbf{A} .
8. Escreva uma função que receba duas matrizes \mathbf{A} e \mathbf{B} e um valor k como parâmetros, e que copie os valores de \mathbf{B} iguais a k para \mathbf{A} .
9. Escreva uma função que receba uma matriz $\mathbf{A}_{m \times n}$, um vetor \mathbf{V} e seu tamanho nulo passado por referência, que copie todos valores de \mathbf{A} para \mathbf{V} e retorne \mathbf{V} . Após a execução da função o vetor terá seu tamanho igual a $n.m$. (Note que o vetor será alocado dinamicamente na função.)
10. Escreva uma nova versão da função do exercício anterior para copiar todos os valores do vetor \mathbf{V} (já povoado) para a matriz \mathbf{A} .
11. Escreva uma função que receba três vetores \mathbf{A} , \mathbf{B} e \mathbf{U} de inteiros, representando conjuntos, e seus respectivos tamanhos, e processe a operação de **união** $\mathbf{U} = \mathbf{A} \cup \mathbf{B}$. (Você deverá criar o vetor \mathbf{U} , dinamicamente, antes de chamar a função.)
12. Escreva uma função que receba três vetores \mathbf{A} , \mathbf{B} e \mathbf{I} de inteiros, representando conjuntos, e seus respectivos tamanhos, e processe a operação de **interseção** $\mathbf{I} = \mathbf{A} \cap \mathbf{B}$. (Você deverá criar o vetor \mathbf{I} , dinamicamente, antes de chamar a função.)
13. 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 de dado. O programa deve copiar todos os valores de \mathbf{A} para \mathbf{B} , na ordem inversa, ou seja, $a_{11} = b_{mn}, a_{12} = b_{m-1n}$, etc, até que $a_{mn} = b_{11}$. Utilize a técnica de *refinamentos sucessivos* quebrando, sucessivamente, o problema em subproblemas, até que seja possível escrever pequenas funções para resolvê-los.
14. Escreva um programa que leia uma matriz de inteiros e um valor x . O programa deve realizar uma busca na matriz à procura da primeira ocorrência de x . Se x for encontrado, imprima sua posição (i, j) . Se x não for encontrado, emita uma mensagem apropriada para este caso. Utilize a técnica de *refinamentos sucessivos* quebrando, sucessivamente, o problema em subproblemas, até que seja possível escrever pequenas funções para resolvê-los.
15. Escreva um função que receba uma matriz $\mathbf{A}_{m \times n}$, e retorne a média de todos os elementos de \mathbf{A} . Após a chamada da função, imprima a matriz em formato matricial e a média.

16. Escreva uma função que receba três matrizes reais $\mathbf{A}_{m \times n}$, $\mathbf{B}_{n \times q}$ e $\mathbf{P}_{m \times q}$. A função deve gerar a **matriz produto** $\mathbf{P}_{m \times q} = \mathbf{A}_{m \times n} \cdot \mathbf{B}_{n \times q}$. Lembre-se: $(A \cdot B)_{ij} = \sum_{k=1}^n a_{ik}b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$.

17. Escreva uma função que receba uma matriz inteira e um inteiro k , e que multiplique todos os elementos da matriz por k .

18. Escreva uma função que receba uma matriz $\mathbf{A}_{m \times n}$ e outra $\mathbf{B}_{n \times m}$ e que determine a matriz $\mathbf{B} = \mathbf{A}^T$, a matriz **transposta** de A . A transposta de uma matriz A possui como linhas as colunas correspondentes da matriz A .

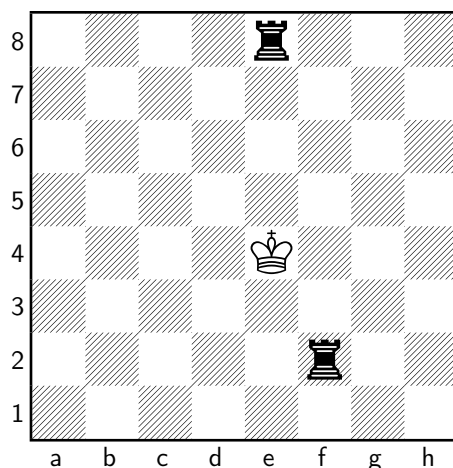
19. Escreva uma função que receba uma matriz inteira \mathbf{A} e um inteiro k . A função deve anular todo o elemento cujo valor seja igual a k .

20. Escreva uma função que receba uma matriz inteira \mathbf{A} e um vetor \mathbf{V} , e que coloque em \mathbf{V} , todos os elementos de \mathbf{A} acima da diagonal principal. Esta função precisa de mais parâmetros?

21. Considere uma matriz $\mathbf{X}_{8 \times 8}$ que represente 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;

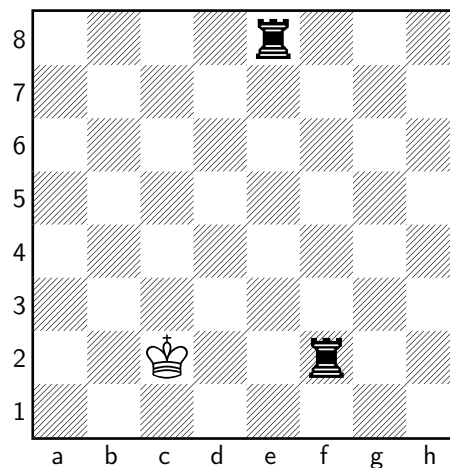
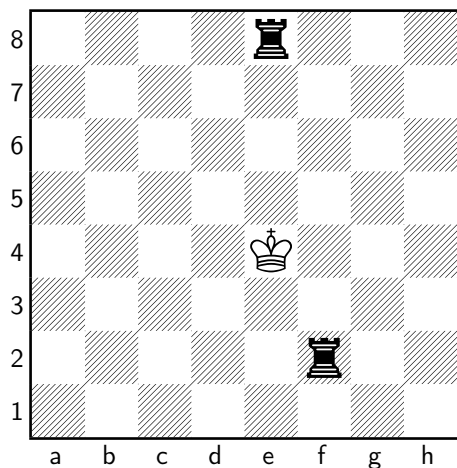
Inicialize um tabuleiro de xadrez com um rei branco e duas torres negras em posições distintas e arbitrárias. Crie uma função que mostre o tabuleiro com suas peças na tela. A primeira figura, abaixo, mostra um tabuleiro de xadrez com o rei branco e as torres negras, e a segunda figura mostra o deveria ser apresentando na tela.



```

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

22. Para o problema do xadrez, considere que a posição (i, j) ($1 \leq i \leq 8$ e $1 \leq j \leq 8$) de uma peça do jogo seja representada por um pequeno vetor de inteiros de duas posições: a posição 0 (zero) do vetor armazena o valor de i e a posição 1 do vetor armazena o valor de j . Escreva uma função que receba as posições do rei branco e de duas torres negras, e 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 mesma linha e/ou coluna em que o rei está. 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.

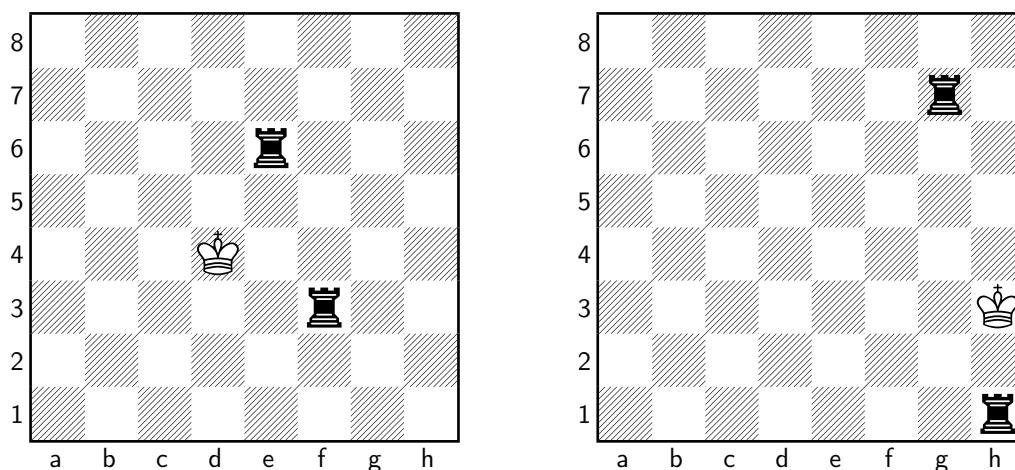


Se o rei não estiver sob ameaça, mostre uma mensagem para indicar esta situação. Faça a função de tal forma que se as peças forem mudadas de posição, a função continuará funcionando

sem a necessidade de modificações. (Note que você não precisa do tabuleiro da questão anterior para resolver o problema.)

23. Crie uma nova versão da função que detecta ameaças sobre o rei, no exercício do xadrez, para que as posições das torres sejam descobertas pelo programa, antes da verificação de ameaça, eliminando a passagem das posições das torres como parâmetros da função.

24. Crie uma nova versão da função do 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. 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.



(Note que você precisa passar como parâmetros o tabuleiro e a posição do rei.)

25. (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 uma função que receba uma matriz de tamanho 3×3 de inteiros distintos (de 1 a 9) e, ao final, e que retorne *true* ou *false*, respectivamente, se a matriz for ou não um quadrado mágico.

26. Um **teletransportador** é uma máquina que transporta uma pessoa de um lugar para outro em uma fração de segundos. O teletransporte funciona da seguinte maneira: os passageiros entram em um teletransportador origem **O**, são desintegrados e reintegrados instantaneamente em um teletransportador destino **D**, localizado em outro lugar do espaço (veja no exemplo abaixo, o que ocorre antes e depois do teletransporte). Escreva uma função que receba duas matrizes

representando os teletransportes **O** e **D**, de ordem $2 \times n$, onde $n = 100$ é uma constante. Cada teletransportador (matriz) deve armazenar n passageiros, e possui duas linhas: a linha 0 armazena os códigos genéticos (inteiros positivos) de cada passageiro; a linha 1 armazena os códigos de cada espécie. Considere os seguintes códigos de espécies:









1 para humanos .

2 para pássaros .









3 para peixes .

Considere que o teletransportador **O** já estará inicializado com n passageiros destas três espécies, misturados, cada um com seu respectivo código genético. Sua função deverá copiar cada passageiro do teletransportador **O** para o teletransportador **D** agrupando os passageiros em **D** por espécie, em ordem crescente de espécie, e ao mesmo tempo, zerar todos os dados do teletransportador **O**.

Antes do teletransporte:

Teletransportador O								Teletransportador D							
2	3	7	5	6	4	1	8	0	0	0	0	0	0	0	0
								0	0	0	0	0	0	0	0

Após o teletransporte:

Teletransportador O								Teletransportador D							
0	0	0	0	0	0	0	0	2	7	8	3	5	4	6	1
0	0	0	0	0	0	0	0								

27. (Baseado na Prova EST/2011-1) Faça uma função que receba uma matriz 100×100 de inteiros, e que coloque em um vetor **PARES**, todos os valores pares encontrados nas linhas ímpares, marcando estes valores na matriz com 2, e que coloque em um vetor **IMPARES**, todos os valores ímpares encontrados nas linhas pares, marcando estes valores na matriz com 1. Após a chamada da função, imprima a matriz e os vetores (somente com as posições preenchidas dos vetores).

28. Escreva uma função que receba duas *strings* de tamanhos iguais e verifique se estas são **anagramas**. Um anagrama é um arranjo das letras de uma palavra (ou frase) para produzir outras palavras, utilizando todas as letras originais exatamente uma vez. Diferente dos palíndromos, o arranjo não precisa ser igual, pois todo palíndromo é um anagrama, mas nem todo anagrama é um palíndromo. Exemplos de anagrama: *Celia* e *Alice*, e *Lyseu* e *Suely*. A função deve retornar um valor lógico.

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

funções com parâmetros para resolver o problema.

30. 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 uma função para compactar uma (*string*) de entrada, gerando um vetor de inteiros de saída que representará uma versão compactada da *string* de entrada. Utilize o seguinte algoritmo de compactação:

- 1) Tome o primeiro caracter α da *string* de entrada.
- 2) Verifique cada posição onde α ocorre na *string* de entrada, e armazene no vetor de saída a seguinte sequência de valores: o número 0 (zero), seguido do valor do caracter α convertido em inteiro, seguido das posições de ocorrência de α . 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 α em um vetor temporário de caracteres (mesmo tamanho do vetor de entrada).
- 4) Pegue o próximo caracter α 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 α 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.

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

31. Escreva uma função para descompactar uma *string* compactada pela função do problema anterior. A função deve tomar o vetor compactado como entrada e gerar o texto original como saída. Utilize uma lógica inversa à do algoritmo descrito no problema anterior.

32. Faça uma função que receba um vetor **A** de inteiros, e um inteiro k pertencente ao vetor. O programa deve particionar **A** em três porções: a primeira contendo todos os elementos menores do que k ; a segunda com k ; e a terceira, com todos os elementos maiores do que k . Não utilize outro vetor como auxiliar. Exemplo: Entrada: **A** = [4, 7, 2, 3, 6, 1, **5**] e $k = 5$. Saída: **A** = [4, 2, 3, **5**, 7, 6].

33. 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. Utilize funções para resolver o problema.

34. Faça uma função para gerar 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

35. Faça uma função para gerar 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