

---

**Introdução à Ciência da Computação**  
**Lista 9 – Vetores, Matrizes e Strings**  
**Prof. Dr. Rogério Galante Negri**

---

1. Dada uma sequência de  $n$  números, imprimi-la na ordem inversa à da leitura.
2. Implemente uma função, com uso de vetores, que determina o número de ocorrência de cada face de um dado em uma simulação de  $n$  lançamentos.
3. Dadas as coordenadas reais  $x$  e  $y$  de um ponto, um número natural  $n$ , e as coordenadas reais de  $n$  pontos ( $1 < n < 100$ ). Deseja-se calcular e imprimir sem repetição os raios das circunferências centradas no ponto  $(x, y)$  que passam por pelo menos um dos  $n$  pontos dados. Implemente um programa que resolva este problema. Use a distância euclidiana como métrica e considere que pontos estão na mesma circunferência se estão à mesma distância do centro.
4. Dada um vetor  $n$ -dimensional, implemente uma função que retorne outro vetor cujos elementos correspondem às posições que define uma ordem crescente dos valores do vetor original. Por exemplo, dado  $\mathbf{v} = [1, 4, 0, 5, 5, 2]$ , o vetor que define a ordem crescente será  $\mathbf{w} = [2, 0, 5, 1, 3, 4]$ . No caso do número 5 que aparece duas vezes em  $\mathbf{v}$ , a ordem dos índices deve-se aparecer em ordem crescente.
5. Dada uma sequência de  $n$  números reais, determinar os números que compõem a sequência e o número de vezes que cada um deles ocorre na mesma.
6. Dados dois números naturais  $m$  e  $n$  e duas sequências com  $m$  e  $n$  números inteiros, obter uma única sequência ordenada contendo todos os elementos das sequências originais sem repetição.
7. Dada uma sequência de  $n$  números inteiros, determinar um segmento de soma máxima. Por exemplo, em 5, 2, -2, -7, 3, 14, 10, -3, 9, -6, 4, 1, a soma do segmento é 33.
8. Dado  $\mathbf{A} \in \mathbb{R}^{m \times n}$  e  $\mathbf{v} \in \mathbb{R}^n$ , escreva um programa que computa o produto de  $\mathbf{A}$  por  $\mathbf{v}$ .

- 
9. Dadas duas matrizes reais  $\mathbf{A} \in \mathbb{R}^{m \times n}$  e  $\mathbf{B} \in \mathbb{R}^{n \times p}$ , implemente uma função que retorna a multiplicação de  $\mathbf{A}$  por  $\mathbf{B}$ .
  10. Escreva um programa que imprima as  $n \leq 20$  primeiras linhas do Triângulo de Pascal. Use uma matriz para armazenar a cálculo do triângulo e em seguida acesse-a para imprimir os valores.
  11. Imprimir as  $n$  primeiras linhas do triângulo de Pascal usando apenas um vetor.
  12. Os elementos  $a_{ij}$  de uma matriz inteira  $\mathbf{A} \in \mathbb{R}^{n \times n}$  representam os custos de transporte da cidade  $i$  para a cidade  $j$ . Dados  $n$  itinerários, cada um com  $k$  cidades, calcular o custo total para cada itinerário. Por exemplo, seja  $\mathbf{A} = \begin{pmatrix} 4 & 1 & 2 & 3 \\ 5 & 2 & 1 & 400 \\ 2 & 1 & 3 & 8 \\ 7 & 1 & 2 & 5 \end{pmatrix}$ , o custo do itinerário 0 3 1 3 3 2 1 0 é  $a_{03} + a_{31} + a_{13} + a_{33} + a_{32} + a_{21} + a_{10} = 3 + 1 + 400 + 5 + 2 + 1 + 5 = 417$
  13. Um conjunto pode ser representado por um vetor da seguinte forma:  $V[0]$  é o tamanho do conjunto;  $V[1]$ ,  $V[2]$ , ... são os elementos do conjunto (sem repetições).
    - a) Faça uma função chamada `interSets` que dados dois conjuntos de números inteiros  $\mathbf{A}$  e  $\mathbf{B}$ , constrói um terceiro conjunto  $\mathbf{C}$  que é a intersecção de  $\mathbf{A}$  e  $\mathbf{B}$ . Lembre-se de que em  $\mathbf{C}[0]$  a sua função deve colocar o tamanho da intersecção;
    - b) Faça uma função chamada `unionSets` que dados dois conjuntos de números inteiros  $\mathbf{A}$  e  $\mathbf{B}$ , constrói um terceiro conjunto  $\mathbf{C}$  que é a união de  $\mathbf{A}$  e  $\mathbf{B}$ .
  14.
    - a) Escreva uma função que recebe  $\mathbf{A} \in \mathbb{R}^{m \times n}$  e determina a sua transposta;
    - b) Dada  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , calcule o produto  $\mathbf{A} \cdot \mathbf{A}^T$ ;
    - c) Faça uma função que verifica se  $\mathbf{A} \in \mathbb{R}^{n \times n}$  é uma matriz identidade.
  15. Dados duas strings (um contendo uma frase e outro contendo uma palavra), determine o número de vezes que a palavra ocorre na frase. Por exemplo, a palavra “ANA” ocorre 5 vezes em “ANA PÉ-DE-CANA É BACANA BEBEDO VODKA DE BANANA”. Escreva um programa que realize tal verificação.

16. Faça uma pesquisa sobre as seguintes funções:

- a) `strlen( )` que retorna o número de caracteres da string
- b) `strcat( )` que concatena duas strings
- c) `strcmp( )` que compara duas strings
- d) `strcpy( )` que copia o conteúdo de uma variável `char[]` para outra `char[]`
- e) `strncat( )` que concatena duas strings (#cars. fixo)
- f) `strncmp( )` que compara duas strings (#cars. fixo)
- g) `strncpy( )` que copia o conteúdo de uma variável `char[]` para outra `char[]` (#cars. fixo)

17. Implemente:

- a) Escreva a função de protótipo `void strlwr(char[])` que converta uma string em letras minúsculas
- b) Escreva a função de protótipo `int strchr(char[], char)` que procura a primeira ocorrência do caractere `ch` em `str`. Tal função retorna o índice do caractere, se for encontrado, ou `-1` caso contrário
- c) Escreva uma função de protótipo `void strinv(char[])` que inverta os caracteres de uma string
- d) Escreva uma função de protótipo `void strins(char[], char, int)` que insira o caractere `ch` na posição `pos` da string `str`
- e) Escreva uma função de protótipo `void strcatn(char[], char[], int)` que copie os `n` últimos caracteres da primeira string na segunda. Use `strncpy( )`.
- f) Escreva a função de protótipo `void strupr(char[])` que converta uma string em letras maiúsculas