

2ª Lista de Exercícios AEDS1

- 1. Crie um vetor de 100 elementos com valores aleatórios de 1 a 500. Elimine todos os valores duplicados e exiba o vetor final em ordem decrescente.**
- 2. Faça um programa que leia 50 números e informe a maior sequência crescente contínua dentro do vetor.**
- 3. Dado um vetor de 30 inteiros, crie um novo vetor contendo apenas os valores que são múltiplos de 3 ou 7, sem repetição.**
- 4. Receba um vetor de 60 números inteiros e separe os valores em 3 vetores distintos: positivos, negativos e zeros. Exiba todos.**
- 5. Leia dois vetores de 15 posições e gere um terceiro vetor com os elementos que estão presentes em ambos, sem duplicatas.**
- 6. Leia uma matriz 6x6 de inteiros e troque a primeira coluna com a última, a segunda com a penúltima e assim por diante.**
- 7. Verifique se uma matriz 5x5 é simétrica, se sua diagonal principal está ordenada e se os elementos abaixo dela são todos negativos.**
- 8. Dada uma matriz 8x8 com números inteiros, calcule a soma dos elementos de cada linha e armazene em um vetor. Em seguida, ordene o vetor.**
- 9. Crie uma matriz 9x9 que simule a movimentação de um cavalo no xadrez. Dado um ponto inicial, marque todas as posições que ele pode atingir.**
- 10. Leia uma matriz 7x7 e substitua cada elemento pela diferença entre ele e a média de seus vizinhos (direita, esquerda, cima e baixo).**
- 11. Escreva uma função que receba uma string e substitua cada consoante por '#' e cada vogal pelo número de sua posição na palavra.**
- 12. Leia uma frase e indique quantas palavras possuem todas as letras distintas entre si (sem letras repetidas).**

13. Leia duas strings e verifique se uma é rotação da outra (ex: "abcd" e "cdab" são rotações).
14. Receba uma string contendo números separados por vírgulas (ex: "12,45,78"). Extraia os números e calcule a média.
15. Leia uma string e verifique se ela contém somente letras minúsculas e está em ordem alfabética crescente.
16. Implemente uma função que receba um vetor de inteiros e retorne a mediana dos valores.
17. Crie uma função que receba duas palavras e retorne o número de letras que estão na mesma posição e são iguais.
18. Desenvolva uma função que receba uma matriz $N \times N$ e retorne verdadeiro se todas as diagonais secundárias forem formadas por múltiplos de 5.
19. Implemente uma função que receba uma string e uma letra, e conte quantas vezes essa letra aparece, ignorando maiúsculas/minúsculas.
20. Escreva uma função que simule o comportamento de um split: receba uma string e um delimitador, retornando um vetor de palavras.
21. Crie um programa que leia uma frase, divida as palavras em um vetor e ordene apenas as que começam com vogal, mantendo as demais na posição original.
22. Leia 3 vetores de nomes (até 20 nomes cada) e armazene em uma matriz. Crie um relatório com os nomes que aparecem nos três vetores.
23. Implemente um jogo da velha onde a matriz de jogadas é armazenada em uma função, e as jogadas são validadas por meio de strings de posição (ex: "A1").
24. Faça um programa que leia um texto e gere uma matriz de frequência, indicando quantas vezes cada letra (a-z) aparece em cada parágrafo.

25. Leia uma matriz 5x5 de strings e verifique se há alguma palavra de 3 letras que se repete em mais de uma linha.
26. Crie um programa que leia um vetor de 50 elementos e gere uma matriz $n \times n$ com os elementos em espiral, chamando funções auxiliares para isso.
27. Desenvolva um sistema de login com verificação de senha forte, onde a senha é validada com uma função e o resultado armazenado em um vetor de status.
28. Faça um programa que leia 10 frases, armazene em uma matriz de caracteres e gere um vetor com o número de palavras únicas por frase.
29. Implemente um verificador de expressões matemáticas simples com vetores de operandos e operadores armazenados separadamente.
30. Simule uma fila circular de atendimento com vetor de structs (nome, idade, prioridade). Crie funções para inserir, remover e reordenar por prioridade.