

Exercícios GEMINI

Exercício 1: Soma de Elementos em um Array 1D

Objetivo: Praticar a criação e iteração sobre um array unidimensional com um loop `for`.

Enunciado: Crie um programa que declare um array de inteiros (`int[]`) com 5 elementos. Preencha o array com quaisquer valores. Em seguida, use um loop `for` para calcular e exibir a soma de todos os elementos do array.

Exercício 2: Encontrando o Maior Valor em um Array 1D

Objetivo: Aprimorar o uso de loops e condicionais com arrays.

Enunciado: Escreva um programa que declare um array de números de ponto flutuante (`double[]`) e o inicialize com alguns valores. Utilize um loop `foreach` para percorrer o array e encontrar o maior valor presente nele. Ao final, exiba o maior valor encontrado.

Exercício 3: Invertendo uma String

Objetivo: Praticar a manipulação de strings e o acesso a caracteres individuais.

Enunciado: Crie um programa que solicite ao usuário que digite uma palavra. Em seguida, leia a palavra e, usando um loop `for`, crie e exiba a string invertida. Por exemplo, se a entrada for "hello", a saída deverá ser "olleh".

Exercício 4: Contagem de Vogais

Objetivo: Trabalhar com strings, loops e condicionais.

Enunciado: Desenvolva um programa que peça ao usuário para inserir uma frase. O programa deve contar e exibir o número total de vogais (a, e, i, o, u), desconsiderando se são maiúsculas ou minúsculas.

Dica: Converta a string inteira para minúsculas antes de começar a contagem para simplificar a lógica.

Exercício 5: Tabela de Multiplicação (Matriz 2D)

Objetivo: Entender a criação e o preenchimento de um array bidimensional (matriz).

Enunciado: Crie um programa que declare uma matriz 10x10 de inteiros (`int[,]`). Preencha esta matriz de forma que o elemento na posição `[i, j]` seja o resultado de $(i + 1) * (j + 1)$. Em seguida, use loops aninhados para imprimir a matriz formatada no console, exibindo a tabela de multiplicação do 1 ao 10.

Exercício 6: Média dos Alunos (Matriz 2D)

Objetivo: Praticar o manuseio de dados em uma matriz e cálculos básicos.

Enunciado: Escreva um programa que armazene as notas de 4 alunos em 3 provas numa matriz 4x3 de `double[,]`.

1. Peça para o usuário digitar as notas de cada aluno.
 2. Calcule e exiba a média de cada aluno (a média das linhas).
 3. Calcule e exiba a média de cada prova (a média das colunas).
-

Exercício 7: Palíndromo

Objetivo: Combinar manipulação de strings e lógica de comparação.

Enunciado: Crie um programa que verifique se uma palavra ou frase fornecida pelo usuário é um palíndromo. Um palíndromo é uma palavra que pode ser lida da mesma forma da esquerda para a direita e da direita para a esquerda. Ignore espaços e diferenças entre maiúsculas e minúsculas.

Exemplos: "arara", "ovo", "Anotaram a data da maratona".

Exercício 8: Soma das Diagonais de uma Matriz

Objetivo: Aprofundar o trabalho com matrizes e lógica de acesso a índices específicos.

Enunciado: Desenvolva um programa que crie uma matriz quadrada 4x4 de inteiros (`int[,]`) e a preencha com valores numéricos. Em seguida, calcule e exiba a soma dos elementos da **diagonal principal** e da **diagonal secundária**.

Exercício 9: Separando Palavras de uma Frase

Objetivo: Utilizar métodos de string e arrays para processamento de texto.

Enunciado: Escreva um programa que solicite uma frase ao usuário. Utilize o método `Split()` da classe `string` para separar a frase em palavras individuais, armazenando-as em um array de strings (`string[]`). Por fim, use um loop `foreach` para exibir cada palavra em uma nova linha.

Exercício 10: Jogo da Velha (Verificação de Vencedor)

Objetivo: Aplicar lógica complexa em uma matriz 2D.

Enunciado: Crie um programa que represente um tabuleiro de Jogo da Velha usando uma matriz 3x3 de caracteres (`char[,]`). A matriz pode ser preenchida com 'X', 'O' ou um caractere de espaço ' ' para representar uma casa vazia. O programa não precisa ser um jogo completo; basta preencher a matriz com uma configuração de jogo finalizada.

O desafio é criar uma lógica para verificar e anunciar o vencedor, analisando:

- Todas as linhas.
- Todas as colunas.
- As duas diagonais.

Se encontrar um vencedor, exiba "O jogador 'X' venceu!" ou "O jogador 'O' venceu!". Se o tabuleiro estiver completo e não houver vencedor, exiba "Deu velha!".

EXERCICIOS CHATGPT

♦ Exercício 1: Soma dos Elementos de um Array

Crie um array de inteiros com 10 posições. Preencha com valores fixos e use um `for` para calcular a soma de todos os elementos.

♦ Exercício 2: Contar Palavras com determinada letra

Peça para o usuário digitar 5 palavras. Armazene em um array de strings e, usando `foreach`, exiba quantas palavras começam com a letra "A" (maiúscula ou minúscula).

♦ Exercício 3: Média dos Números Pares

Crie um array de 20 números inteiros aleatórios (entre 1 e 100). Usando `for` ou `foreach`, calcule e exiba a **média apenas dos números pares**.

♦ Exercício 4: Matriz de Notas

Crie uma matriz 2D de notas (4 alunos x 3 provas). Peça os dados via `Console.ReadLine()` e depois mostre:

- A média de cada aluno
 - Qual aluno teve a maior média
-

♦ Exercício 5: Inverter Palavras

Peça uma palavra ao usuário e mostre a mesma palavra invertida (ex: "casa" → "asac"). Use manipulação de `char[]` ou métodos da classe `string`.

♦ Exercício 6: Tabela de Multiplicação

Usando uma matriz 10x10, preencha com a tabuada do 1 ao 10 (ex: linha 2, coluna 3 = 2x3 = 6). Depois, exiba a matriz formatada.

♦ Exercício 7: Encontrar Maior e Menor Palavra

Peça para o usuário digitar uma frase. Separe as palavras com `.Split(' ')` e diga:

- Qual é a maior palavra

- Qual é a menor palavra
-

♦ Exercício 8: Substituição de Caracteres

Peça ao usuário uma frase e substitua todas as vogais por *. Ex: "Olá mundo" → "/ mnd". Use `foreach` e `string.Concat` para reconstruir a frase.

♦ Exercício 9: Soma das Diagonais de uma Matriz

Crie uma matriz quadrada 3x3 com valores fixos. Calcule a soma da diagonal principal e da secundária. Exiba ambos os resultados.

♦ Exercício 10: Verificar Palíndromo

Peça uma palavra ao usuário e diga se ela é um palíndromo (ex: "radar", "osso"). Ignore maiúsculas/minúsculas. Use `ToLower()`, `Reverse()`, etc.