

**01.** Crie uma função que recebe um array de 2 números e coloque eles em ordem crescente. Não use o método sort.

**02.** Crie uma função que recebe um array de 3 números e coloque eles em ordem crescente. Crie sua própria função para isso.

**03.** Crie um script que pergunte 3 números ao usuário, coloque em um array, depois exiba tal array. Modifique os elementos do array de modo a a sequência de números ficar do contrário.

Ou seja, se digitou: 1,2,3

Vai aparecer: 3,2,1

**04.** Refaça o script anterior, mas agora pergunte ao usuário quantos números ele deseja inserir no array.

Depois, o usuário vai preenchendo elemento por elemento do vetor.

Ao final, exiba o array e o array ao contrário (generalização do script anterior).

**05.** Faça uma função que recebe um array de qualquer tamanho e ordene seus elementos na ordem crescente. Faça sua função na mão, não invoque nenhum método de *sorting*. Dica: pesquise sobre *bubble sort*.

**06.** Crie uma função que recebe um vetor e um número.

Ela deve mostrar todos os índices onde esse número aparece no vetor, e não só apenas o primeiro e/ou último índice em que o número aparece (como nos métodos indexOf e lastIndexOf).

**07.** Crie um script que pede um inteiro positivo para o usuário. Em seguida, popule uma array com os números de Fibonacci, na hora de somar com os termos anteriores, some com elementos anteriores do array.

**08.** Faça o mesmo do exercício anterior, mas com fatorial.

Fatorial de 0:  $0! = 1$

Fatorial de 1:  $1 \times \text{elemento0} = 1$

Fatorial de 2:  $2 \times \text{elemento1} = 2$

Fatorial de 3:  $3 \times \text{elemento2} = 3 \times 2 = 6$

...

Fatorial de n:  $n \times \text{elemento}(n-1) = n \times (n-1)! = n!$

**08.** Usando matriz (array de array), crie o jogo da velha. Inicialmente, as posições são variáveis vazias.

O usuário fornece a linha e a coluna e clica num botão, então é marcado no tabuleiro X ou O (se for o jogador 1 ou jogador 2) que

**09.** Peça ao usuário uma quantidade de linhas e e outra de colunas, depois declare tal matriz, inicialize todos os elementos de 1 até 'linha x coluna'. **10.** Crie uma função que receba um array de números e retorne o **maior valor**, sem usar Math.max.

Perfeito, agora entendi 👍

Você quer **somente exercícios, sem dividir por “arrays + strings”, “matrizes” etc, sem soluções**, escritos **no mesmo estilo simples e direto** da lista que você enviou no começo, **com explicações claras do que deve ser feito**.

Abaixo estão **mais 19 exercícios**, continuando a numeração, todos **sobre arrays/vetores e matrizes**, focados em **lógica**, do jeito que normalmente professores passam.

**10.**

Crie uma função que recebe um array de números e retorna o **maior número** do array.  
Não use Math.max.

**11.**

Crie uma função que recebe um array de números e retorna o **menor número** do array.  
Não use Math.min.

**12.**

Crie uma função que recebe um array de números e retorna a **soma de todos os elementos**.

**13.**

Crie uma função que recebe um array de números e retorna a **média aritmética** desses valores.

**14.**

Crie uma função que recebe um array de números e retorna um **novo array apenas com os números pares**.

**15.**

Crie uma função que recebe um array de números e retorna um **novo array apenas com os números ímpares**.

**16.**

Crie uma função que recebe um array e um número.  
A função deve informar **quantas vezes esse número aparece** dentro do array.

**17.**

Crie uma função que recebe um array e retorna outro array **sem valores repetidos**.

Exemplo:

Entrada → [1, 2, 2, 3, 4, 4]

Saída → [1, 2, 3, 4]

**18.**

Crie um script que peça **5 números ao usuário**, armazene em um array e exiba:

- o array original
- o array invertido (sem usar reverse)

**19.**

Crie uma função que recebe dois arrays e retorna um **terceiro array com todos os elementos dos dois**.

**20.**

Crie um script que peça ao usuário um número **N**.  
Depois, crie um array com os números de **1 até N** e exiba esse array.

**21.**

Crie uma função que recebe um array de números e verifica se ele está **em ordem crescente**. Retorne true se estiver ordenado e false caso contrário.

**22.**

Crie uma função que recebe um array e retorna um **novo array com cada elemento multiplicado por 2**.

**23.**

Crie um script que peça ao usuário a quantidade de elementos do array. Depois, o usuário preenche o array e, ao final, exiba:

- o array
- a soma dos valores
- a média

**24.**

Crie uma função que recebe um array e retorna um **novo array ordenado em ordem decrescente**, sem usar sort.

**25.**

Crie uma matriz 3x3 e preencha todos os valores manualmente. Depois, exiba a matriz linha por linha no console.

**26.**

Crie uma função que recebe uma matriz e retorna a **soma de todos os elementos**.

**27.**

Crie um script que peça ao usuário o número de linhas e colunas de uma matriz. Depois, preencha essa matriz com números informados pelo usuário e exiba o resultado.

**28.**

Crie uma função que recebe uma matriz e retorna um array contendo a **soma de cada linha** da matriz.

**29.**

Crie um script que simule um **jogo da velha**, usando matriz. O usuário informa linha e coluna, e o programa alterna entre jogador X e jogador O.

**30.**

Crie uma função que recebe um array de números e retorna um **novo array com os valores positivos**.