

## Lista de Exercícios 08

### Aula 09

Estrutura de Repetição: enquanto, faça-enquanto, para

Estrutura de Dados: vetor

Biblioteca: Util

#### >> Exercícios de Aprofundamento<<

##### 1) Maior, menor e posições

Crie um algoritmo que leia 10 valores do usuário. Informe qual o maior e sua posição e qual o menor e sua posição.

##### 2) Preenchimento alternado

Desenvolva um algoritmo com dois vetores de 10 posições cada, de números inteiros. Receba do usuário 20 números. À medida que vá lendo, preencha os vetores alternadamente: O primeiro valor no vetor 1, o segundo coloque no vetor 2, o terceiro no vetor 1...

No final, exiba os valores no vetor 1 e vetor 2.

##### 3) Mais alto e mais baixo 2

Crie um programa com dois vetores, cada um com 6 posições, um com os nomes dos estudantes, outro com suas respectivas alturas. **Receba o nome e a altura de cada estudante.** Ao final, exiba qual o estudante mais alto e o mais baixo da turma e sua posição no vetor.

##### 4) Média da Turma

Receba do usuário o nome de cada um dos 6 estudantes de uma turma. Em seguida, receba do usuário as notas de cada estudantes em uma atividade, seguindo o exemplo:

Insira a nota do estudante [nome do estudante] na atividade 01:

O próximo passo é receber a nota de uma segunda atividade, de cada estudante:

Insira a nota do estudante [nome do estudante] na atividade 02:

Por fim, exiba o nome de cada estudante acompanhado de sua média respeitando a formatação:

[nome do estudante]: [média]

**5) Vetor ímpar, vetor par** 🏠🏠🏠

Crie dois vetores com 5 posições, um para números inteiros ímpares e outro com números inteiros pares. Receba do usuário vários números e vá preenchendo os vetores de forma adequada. Encerre o programa apenas quando os dois vetores estiverem preenchidos, exibindo os valores armazenados neles. Se um dos vetores for preenchido antes do término do programa, não insira mais valores nele. Obs.: Se o usuário inserir o valor 0 (zero), exiba a mensagem "Zero não pode ser inserido" e continue o programa.

**6) Tem aqui? 2** 🏠🏠🏠

🧐 Use a função sorteia da biblioteca Util

Crie um programa de adivinhação. Em um vetor que contenha 5 números inteiros **gerados aleatoriamente**, o usuário deve tentar descobrir ao menos um desses números. Receba o palpite do usuário e verifique se o número está na lista. Se estiver, imprima o número que foi encontrado e diga ao usuário a posição no vetor. Caso contrário, informe que o número não pertence a lista e tente novamente.

**7) Divisível pelas entradas** 🏠🏠

Faça um algoritmo que leia e mostre um vetor de 10 números inteiros. A seguir, peça ao usuário para informar um valor inteiro e positivo e mostre todos os números do vetor que são divisíveis por esse número.


**8) Lista de primos** 🏠🏠🏠

Escreva um algoritmo que gera os 10 primeiros números primos a partir de 1 e os armazena em um vetor de 10 posições escrevendo, no final, o vetor.

**9) Aleatoriamente primo** 🏠🏠🏠

🧐 Use a função sorteia da biblioteca Util

Crie um programa que preencha um vetor de 10 posições com números inteiros aleatórios entre 2 e 200. Informe ao usuário quantos dos números neste vetor são primos.

P.S.: Tem mais um exercício aqui 

**10) Aritmética com vetores** 🧑🧑🧑

Receba do usuário 10 números inteiros, em seguida, receba 10 caracteres aritméticos (+, -, \*, /) e então, receba mais 10 números. Exiba, em seguida, linha a linha das operações aritméticas resultantes.

Exemplos Entrada (com vetor de 3 posições)	Exemplos Saída (com vetor de 3 posições)
1 2 3 + - * 4 7 2	1 + 4 = 5 2 - 7 = -5 3 x 2 = 6