**Exercícios de Ponteiros para Vetores**

**1-Faça um programa que peça 10 valores e, depois, apresente-os na tela.**

#include <iostream>

int main() {

int valores[10];

std::cout << "Digite 10 valores:\n";

for (int i = 0; i < 10; i++) {

std::cin >> valores[i];

}

std::cout << "Valores digitados:\n";

for (int i = 0; i < 10; i++) {

std::cout << valores[i] << " ";

}

return 0;

}

**2-Faça um programa que peça 10 valores e, depois, apresente-os na tela de trás para frente.**

#include <iostream>

int main() {

int valores[10];

std::cout << "Digite 10 valores:\n";

for (int i = 0; i < 10; i++) {

std::cin >> valores[i];

}

std::cout << "Valores digitados de trás para frente:\n";

for (int i = 9; i >= 0; i--) {

std::cout << valores[i] << " ";

}

return 0;

}

**3-Faça um programa que peça 10 valores e, depois, apresente o maior valor.**

#include <iostream>

int main() {

int valores[10];

int maior;

std::cout << "Digite 10 valores:\n";

for (int i = 0; i < 10; i++) {

std::cin >> valores[i];

if (i == 0 || valores[i] > maior) {

maior = valores[i];

}

}

std::cout << "Maior valor digitado: " << maior << "\n";

return 0;

}

**4-Faça um programa que peça 10 valores e, depois, apresente-os, mas multiplicado por 2;**

#include <iostream>

int main() {

int valores[10];

std::cout << "Digite 10 valores:\n";

for (int i = 0; i < 10; i++) {

std::cin >> valores[i];

valores[i] \*= 2;

}

std::cout << "Valores digitados, multiplicados por 2:\n";

for (int i = 0; i < 10; i++) {

std::cout << valores[i] << " ";

}

return 0;

}

**5-Faça um programa como o de média de nota, mas com funções.**

#include <iostream>

// Função para calcular a média

float calcularMedia(int notas[], int tamanho) {

int soma = 0;

for (int i = 0; i < tamanho; i++) {

soma += notas[i];

}

return static\_cast<float>(soma) / tamanho;

}

int main() {

int notas[10];

std::cout << "Digite as 10 notas:\n";

for (int i = 0; i < 10; i++) {

std::cin >> notas[i];

}

float media = calcularMedia(notas, 10);

std::cout << "A média das notas é: " << std::fixed << std::setprecision(2) << media << "\n";

return 0;

}