



Técnicas de Programação – Prof. Ademar Virgolino e Lucas Hartmann
Atividades Sobre Matrizes e Ponteiros

OBJETIVOS

Fixar os conhecimentos de criação de funções, passagem de parâmetros, recursão, e outros explorados em Aula 05 – Matrizes, e Aula 06 – Ponteiros.

REQUISITOS

1. Em um programa crie uma matriz de 20 elementos inteiros, e:
 - a) Inicialize com números inteiros em ordem crescente.
 - b) Imprima o conteúdo original da matriz na tela.
 - c) Utilize `std::swap(...)` para inverter a ordem de cada par de elementos.
 - d) Imprima o conteúdo final da matriz na tela e verifique se o resultado está correto.
2. Em um programa crie uma matriz de 20 elementos inteiros e:
 - a) Preencha a matriz com valores aleatórios usando `std::rand()`.
 - b) Imprima o conteúdo original da matriz na tela.
 - c) Imprima na tela o somatório dos elementos.
 - d) Imprima na tela o maior valor.
 - e) Imprima na tela o menor valor.
3. Tomando os programas anteriores como base, crie duas funções auxiliares para matrizes inteiras de qualquer tamanho. Em cada função receba um ponteiro para a matriz e o número de elementos como parâmetros.
 - a) Uma função `randomize_array(...)` que preenche a matriz com valores aleatórios.
 - b) Uma função `print_array(...)` que imprime a matriz na tela.
4. Em um programa:
 - a) Crie uma função `media_matriz(...)` que receba como parâmetros uma matriz de inteiros, e retorne média de seus elementos como um número em ponto flutuante.
 - b) Crie uma função `desvio_matriz(...)` que receba como parâmetros uma matriz de inteiros e a sua média em ponto flutuante (presumida conhecida), e retorne o desvio padrão em ponto flutuante.
 - c) Crie na função `main` uma matriz com 25 elementos inteiros aleatórios, utilize as funções acima para calcular média e desvio padrão, e imprima na tela.
5. Crie e imprima uma matriz com 15 elementos inteiros aleatórios, ordene, e imprima a resposta final. Implemente seu próprio algoritmo de ordenação utilizando laços de repetição e `std::swap(...)` conforme julgar necessário. Explique seu algoritmo e como ele funciona.



6. Em um programa:

a) Defina função `print_inc_print(...)` que:

- Recebe um ponteiro para um valor inteiro.
- Mostra o valor do inteiro.
- Incrementa o inteiro.
- Mostra o valor novamente.

b) Defina na função `main(...)` uma variável inteira `x` com valor conhecido e:

- Chame `print_inc_print(...)` passando o endereço de `x` como argumento.
- Imprima o valor final de `x`.