

# Algoritmos e Programação II

## Lista de Exercícios: Array

Prof. Evandro C. R. Rosa

UNIVALI

Nome Completo: \_\_\_\_\_ Código de Pessoa: \_\_\_\_\_

### 1. Responda sucintamente:

- (a) Observe a seguinte definição de array: `int valores[10];`
  - i. Quantos elementos o array possui?
  - ii. Qual é o índice do primeiro elemento no array?
  - iii. Qual é o índice do último elemento no array?
  - iv. Supondo que um `int` utilize quatro bytes de memória, quanta memória o array utiliza?
- (b) Por que uma função que recebe um array como argumento e o processa também deve aceitar um argumento especificando o tamanho do array?
- (c) Considere a seguinte definição de array: `int valores[5] = { 4, 7, 6, 8, 2 };`  
O que cada uma das seguintes instruções exibe?
  - i. `std::cout << valores[4] << std::endl;`
  - ii. `std::cout << (valores[2] + valores[3]) << std::endl;`
  - iii. `std::cout << ++valores[1] << std::endl;`
- (d) Como você define um array sem fornecer um tamanho na declaração?
- (e) Observe a seguinte definição de array: `int numeros[5] = { 1, 2, 3 };`
  - i. Qual valor está armazenado em `numeros[2]`?
  - ii. Qual valor está armazenado em `numeros[4]`?
- (f) Um array é passado para uma função por valor ou por referência?
- (g) Quando você passa o nome de um array como argumento para uma função, o que é realmente passado?
- (h) Como você estabelece uma relação paralela entre dois ou mais arrays?
- (i) Observe a seguinte definição de array: `double vendas[8][10];`
  - i. Quantas linhas o array possui?
  - ii. Quantas colunas o array possui?
  - iii. Quantos elementos o array possui?
  - iv. Escreva uma instrução que armazena um número na última coluna da última linha do array.
- (j) Ao escrever uma função que aceita um array bidimensional como argumento, qual tamanho deve ser especificado no parâmetro para o array?

2. Os códigos abaixo possuem um ou mais erros. Indique quais são:

- (a) 

```
int tamanho;  
double valores[tamanho];
```
- (b) 

```
int colecao[-20];
```
- (c) 

```
int tabela[10];  
for (int x = 0; x < 20; x++) {  
    std::cout << "Digite o próximo valor: ";  
    std::cin >> tabela[x];  
}
```
- (d) 

```
int horas[3] = 8, 12, 16;
```
- (e) 

```
int numeros[8] = {1, 2, , 4, , 5};
```
- (f) 

```
float avaliacoas[];
```
- (g) 

```
int array1[4], array2[4] = {3, 6, 9, 12};  
array1 = array2;
```
- (h) 

```
void mostrar_valores(int nums) {  
    for (int cont = 0; cont < 8; cont++)  
        std::cout << nums[cont];  
}
```
- (i) 

```
void mostrar_valores(int nums[4][4]) {  
    for (linhas = 0; linhas < 4; linhas++)  
        for (colunas = 0; colunas < 5; colunas++)  
            std::cout << nums[linhas][colunas];  
}
```

3. Escreva um programa que permita ao usuário inserir 10 valores em um array. O programa deve então exibir os maiores e menores valores armazenados no array.

4. Escreva um programa que permita ao usuário inserir a quantidade total de chuva para cada um dos 12 meses do ano em um array de números reais (`double`). O programa deve calcular e exibir a quantidade total de chuva no ano, a média mensal de chuva e os meses com os maiores e menores índices de chuva.

**Validação de Entrada:** Não aceite números negativos para os índices mensais de chuva.

5. Escreva um programa que permita a dois jogadores jogarem uma partida de jogo da velha. Use um array bidimensional de caracteres com três linhas e três colunas como o tabuleiro. Cada elemento do array deve ser inicializado com um asterisco (\*). O programa deve executar um loop que:

- Exibe o conteúdo do tabuleiro (array).
- Permite ao jogador 1 selecionar uma posição no tabuleiro para um X. O programa deve solicitar ao usuário que insira o número da linha e da coluna.
- Permite ao jogador 2 selecionar uma posição no tabuleiro para um O. O programa deve solicitar ao usuário que insira o número da linha e da coluna.
- Determina se um jogador ganhou ou se ocorreu um empate. Se um jogador ganhar, o programa deve declarar esse jogador como vencedor e encerrar. Se ocorrer um empate, o programa deve informar e encerrar.

O jogador 1 vence quando houver três X em uma linha no tabuleiro. Os X podem aparecer em uma linha, em uma coluna ou diagonalmente no tabuleiro. O empate ocorre quando todas as posições do tabuleiro estiverem preenchidas, mas não houver vencedor.

6. Escreva um programa que utilize os seguintes arrays:

- **emp\_id**: um array de sete inteiros para armazenar números de identificação de funcionários. O array deve ser inicializado com os seguintes números:
  - 5658845, 4520125, 7895122, 8777541, 8451277, 1302850, 7580489
- **horas**: um array de sete inteiros para armazenar o número de horas trabalhadas por cada funcionário.
- **valor\_hora**: um array de sete números reais (**double**) para armazenar o valor pago por hora de cada funcionário.
- **salarios**: um array de sete números reais (**double**) para armazenar os salários brutos de cada funcionário.

O programa deve relacionar os dados em cada array através dos índices. Por exemplo, o número no elemento 0 do array **horas** deve ser o número de horas trabalhadas pelo funcionário cujo número de identificação está armazenado no elemento 0 do array **emp\_id**. O valor de pagamento desse mesmo funcionário deve estar armazenada no elemento 0 do array **valor\_hora**.

O programa deve exibir o número de identificação de cada funcionário e solicitar ao usuário que insira as horas trabalhadas e o valor pago por hora desse funcionário. Ele deve então calcular os salários brutos de cada funcionário (horas trabalhadas multiplicadas o valor pago por hora) e armazená-los no array **salarios**. Após os dados terem sido inseridos para todos os funcionários, o programa deve exibir o número de identificação de cada funcionário e seus salários brutos.

**Validação de Entrada:** Não aceite valores negativos para horas trabalhadas ou números menores que 80,00 para o valor pago por hora.

7. Escreva um programa que permita a um fabricante de molhos rastrear as vendas de cinco tipos diferentes de molho: suave, médio, doce, picante e apimentado. O programa deve usar dois arrays paralelos de 5 elementos: um array de strings que armazena os cinco nomes dos molhos e um array de inteiros que armazena o número de potes vendidos durante o mês passado para cada tipo de molho. Os nomes dos molhos devem ser armazenados usando uma lista de inicialização no momento da criação do array de nomes. O programa deve solicitar ao usuário que insira o número de potes vendidos para cada tipo. Após esses dados de vendas terem sido inseridos, o programa deve gerar um relatório que exibe as vendas de cada tipo de molho, as vendas totais e os nomes dos produtos com maior e menor número de vendas.

**Validação de Entrada:** Não aceite valores negativos para o número de potes vendidos.

8. Em um programa, escreva uma função que aceite três argumentos: um array, o tamanho do array e um número **n**. Suponha que o array contenha inteiros. A função deve exibir todos os números no array que sejam maiores que o número **n**. Escreva um programa para testar a função.

9. Um zoológico local deseja acompanhar quantos quilos de comida cada um de seus três macacos consome por dia durante uma semana típica. Escreva um programa que armazene essas informações em um array bidimensional de  $3 \times 5$ , onde cada linha representa um

macaco diferente e cada coluna representa um dia da semana. O programa deve primeiro solicitar ao usuário que insira os dados para cada macaco. Em seguida, ele deve gerar um relatório que inclua as seguintes informações:

- Quantidade média de comida consumida por dia pela família inteira de macacos.
- A menor quantidade de comida consumida durante a semana por qualquer macaco.
- A maior quantidade de comida consumida durante a semana por qualquer macaco.

**Validação de Entrada:** Não aceite números negativos para os quilos de comida consumidos.

10. O Quadrado Mágico de Lo Shu é uma grade com 3 linhas e 3 colunas, mostrado na 1a. O Quadrado Mágico de Lo Shu tem as seguintes propriedades:

- A grade contém os números de 1 a 9 exatamente.
- A soma de cada linha, cada coluna e cada diagonal resulta no mesmo número.

Isso é mostrado na 1b. Em um programa, você pode simular um quadrado mágico usando um array bidimensional. Escreva uma função que aceite um array bidimensional como argumento e determine se o array é um Quadrado Mágico de Lo Shu. Teste a função em um programa.

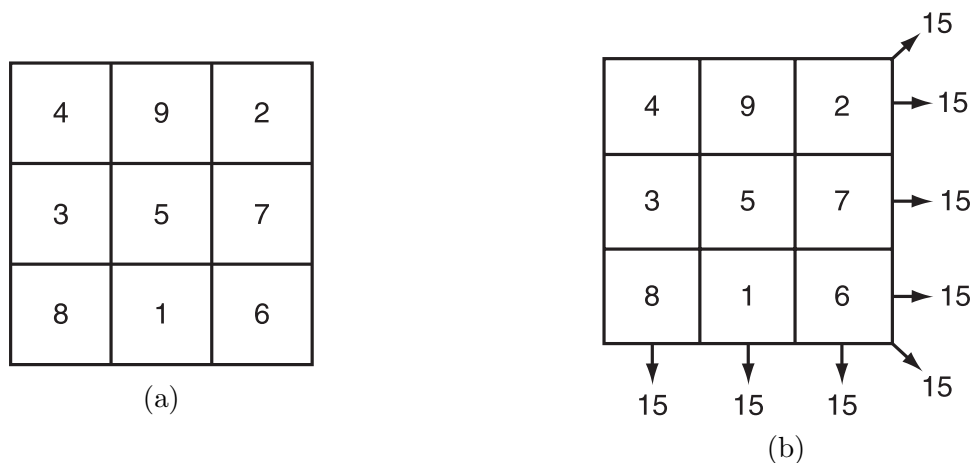


Figura 1: Quadrado Mágico de Lo Shu.