



Universidade Federal da Bahia - UFBA
Instituto de Matemática - IM
Departamento de Ciência da Computação - DCC
Curso de Bacharelado em Ciência da Computação

MATA40 - Estrutura de Dados e Algoritmos

Período: 2020.1

Data: 05/03/2020.

Prof. Antonio L. Apolinário Jr.

Roteiro do Laboratório 1 - Vetores e Matrizes

Objetivos:

- Recordar os conceitos de Agregados Homogeneos unidimensionais (Vetores) e bidimensionais (Matrizes) ;
- Reforçar os conceitos básicos de estruturas de controle em linguagem C;
- Implementar, em linguagem C, programas que manipulam vetores e matrizes.

Conceitos básicos:

Arranjo:

Estrutura de dados que armazena uma coleção de elementos de tal forma que cada um dos elementos pode ser identificado por um ou mais índices. Arranjos unidimensionais são usualmente denominados **vetores**, enquanto que aqueles de dimensão maior ou igual a 2 são chamados de **matrizes**.

Usualmente, todos os elementos do arranjo possuem mesmo tamanho e tipo de dados. Eles podem ser acessados individualmente e instantaneamente por sua posição no arranjo, dada por um índice. O índice geralmente utiliza uma sequência de números inteiros, mas o índice pode ter qualquer valor ordinal. Em linguagem C o índice de um vetor sempre inicia em 0. A seguir um exemplo, em linguagem C, de declaração e acesso aos elementos de um vetor.

```
int a[50]; // Vetor  
a[20] = 20;  
a[44] = a[12] * a[33];
```

No caso de matrizes os processos são análogos, porém envolvem um índice a mais:

```
float m[50][40]; // Vetor  
m[20][10] = -45.3;  
m[1][c] = m[0][1] * m[c][0];
```

Muito cuidado deve ser tomado no controle dos índices para acesso aos elementos dos arranjos! É importante entender que um arranjo é um bloco de memória de endereços contíguos, capaz de armazenar todos os elementos do arranjo. O endereço inicial é armazenado em seu identificador e seus elementos são endereçados a partir de expressões que geram o endereço correspondente dentro do bloco de memória previamente alocado. Essas expressões são:

$\&(A[i]) \rightarrow A + i * \text{sizeof}(\text{tipo de } A)$

$\&(M[i][j]) \rightarrow M + (i * \text{NUM_COL} + j) * \text{sizeof}(\text{tipo de } M)$

Portanto, o compilador calcula um endereço de memória baseado nas informações do arranjo e do índice fornecido, sem criticar se o endereço gerado pertence ou não ao bloco de memória associado ao arranjo. Por isso, cabe ao programador assegurar que esses limites serão respeitados, caso contrário o endereço gerado por causar uma falha de segmentação, um dos erros mais comuns em programação em linguagem C¹.

Roteiro:

1. Um inteiro primo é qualquer inteiro que só pode ser dividido exatamente por si mesmo e por 1. A **Peneira de Eratóstenes** é um método conhecido a muito tempo para encontrar números primos. Ele funciona da seguinte maneira:

- Crie um array com todos os elementos inicializados com verdadeiro.
- Começando com a posição de índice 2 do vetor, para cada elemento i do vetor, se o elemento i do vetor for verdadeiro, percorra o restante do vetor armazenando o valor falso em todos os elementos de índice múltiplos de i .

Ao término desse processo os elementos do vetor com valor verdadeiro representam os números primos, portanto seus índices indicam os números primos.

Escreva um programa que determine e imprima todos os números primos entre 1 e N (fornecido pelo usuário) utilizando o **Algoritmo Crivo de Eratóstenes** tal como descrito acima.

2. Os elementos a_{ij} de uma matriz inteira $A_{n \times n}$ representam os custos de transporte da cidade i para a cidade j . Faça um programa em linguagem C que, dada uma quantidade de cidades n , gere uma matriz de custos com valores aleatórios (entre 0 e 100). Em seguida solicite um itinerário, formado por uma sequência de k cidades e calcule o custo total do itinerário.
3. Considere uma matriz $D_{8 \times 8}$ que representa a posição atual de um jogo de damas, sendo que uma posição com valor 0 indica uma casa vazia, valor 1 indica uma casa ocupada por uma peça branca e -1 indica uma casa ocupada por uma peça preta. Supondo que as peças pretas estão se movendo no sentido crescente das linhas da matriz D , determinar as posições das peças pretas que:
 - podem tomar peças brancas;
 - podem mover-se sem tomar peças;
 - não podem se mover.

A montagem das matrizes dos itens 2 e 3 pode ser feita a partir da leitura de um arquivo texto, ou fornecida pelo usuário interativamente.

¹ Outros problemas comuns:

https://www.cprogramming.com/beginner_programming_mistakes.html
<https://www.cprogramming.com/tutorial/common.html>