



# TÉCNICAS DE PROGRAMAÇÃO

Aula 03 – Vetores e Matrizes



# Objetivos de Aprendizagem

1. Descrever estruturas de dados do tipo vetor e matriz.
2. Desenvolver programas usando vetor e matriz.



# Variáveis Compostas Homogêneas

- Interpretada como um conjunto (coleção) de valores de um mesmo tipo.
- Podem ser
  - Unidimensionais
  - Multidimensionais
- Requerem novos conceitos para serem manipuladas

# Variáveis Compostas Unidimensionais

- São uma coleção caixinhas, onde cada caixinha guarda uma variável.
- Semelhante a uma coleção de gavetas do armário agrupadas.




# Vetores

- Tipo de dado usado para representar uma coleção de variáveis de um mesmo tipo.
- Estrutura de dados homogênea e unidimensional.
- Sintaxe:
  - `tipo nome_do_vetor[tamanho];`
  - `tamanho` representa o número de elementos.
  - O índice do vetor varia de 0 a  $(\text{tamanho} - 1)$

# Vetores

- As variáveis são alocadas sequencialmente na memória, onde o endereço mais baixo corresponde ao primeiro elemento (índice 0) do vetor.



*$x$  é um vetor unidimensional de  $n$  elementos.*

# Exemplo sem o uso de vetores

- Ler a nota de 3 alunos de uma disciplina e calcular a média.

```
float  nota1, nota2, nota3;

printf("Entre com a 1a. nota:");
scanf("%f", &nota1);
printf("Entre com a 2a. nota:");
scanf("%f", &nota2);
printf("Entre com a 3a. nota:");
scanf("%f", &nota3);
printf("Média = %f", (nota1 + nota2 + nota3) / 3);
```

# Exemplo com o uso de vetores

- Ler a nota de 3 alunos de uma disciplina e calcular a média.

```
float  nota[3];
float  soma = 0;
int  i;

for (i = 0; i < 3; i++) {
    printf("Entre com a %da. nota: ", (i + 1));
    scanf("%f", &nota[i]);
    soma = soma + nota[i];
}
printf("Média da disciplina = %f", soma / 3);
```



# Vetores: Inicialização

- A inicialização dos valores atribuídos aos elementos de um vetor pode ser feita no momento da sua declaração, exemplo:

```
main() {  
    int i, vetor[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};  
    for (i = 0; i < 10; i++){  
        printf("%d\n", vetor[i]);  
    }  
}
```

# Vetores

- Em 'C' não existe declaração de vetor dinâmico
- O tamanho de um vetor tem que ser determinado em tempo de compilação.
- Exemplo:

```
int  alunos;  
int  notas [ alunos ];  
  
printf ("entre com o número de alunos");  
scanf ("%d", &alunos);  
  
/* NÃO É ACEITO PELA LINGUAGEM!!! */
```



# Vetores

- C não realiza verificação de limites em vetores
- Nada impede o acesso além do fim do vetor
- Faça sempre que possível a verificação do limite



# Exercício

- Fazer um programa que preenche um vetor de 10 posições contendo as notas dos alunos de uma turma.
- Em seguida o programa deve imprimir a media aritmética dos 10 alunos.

# Exemplo

- Sendo  $V$  o vetor abaixo e as variáveis  $X = 2$  e  $Y = 4$ , quais os valores de:

2	6	8	3	10	9	1	21	33	14
0	1	2	3	4	5	6	7	8	9

- a)  $V[X+1]$
- b)  $V[X] + 1$
- c)  $V[X+Y]$
- d)  $V[8-V[2]]$
- e)  $V[V[V[7]]]$



# Exercício

1. Construa um programa que preenche um vetor de inteiros de 100 números, colocando 0 nas posições par e 1 ímpar.
2. Construa um programa que lê, soma e imprime o resultado da soma de um vetor de inteiros de 10 posições.
3. Construa um programa que multiplique os valores de um vetor de reais de 20 posições pelo valores de um outro vetor de reais de 20 posições. Os resultados das multiplicações devem ser armazenados num terceiro vetor.



# Exercício Desafio

- Construa um programa que leia e guarde os elementos em um vetor de 20 posições.
- Em seguida o algoritmo ordena os elementos do vetor de acordo com a seguinte estratégia:
  - Selecione o elemento do vetor que guarda o menor e o maior valor

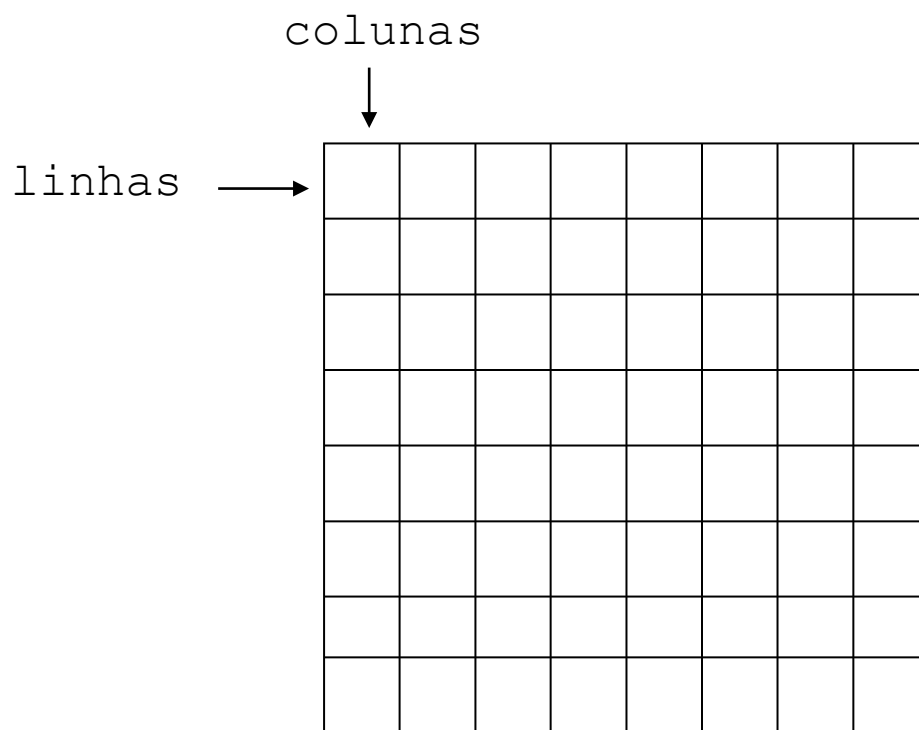
# Variáveis Compostas Bidimensionais

- Precisam de mais de um índice para endereçamento.
- Funciona como um vetor de vetores




# Variáveis Compostas Bidimensionais

- Exemplo:



# Matrizes: Manipulação

MATRIZ

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

MATRIZ[2][3]



# Matrizes

- Em C podemos definir um vetor em que cada posição temos um outro vetor (matriz).
- Matriz é uma estrutura de dados homogênea bidimensional.
- Sintaxe:
  - **tipo nome\_da\_matriz[tamanho1][tamanho2];**
  - **tamanho1** representa o número de linhas da matriz e **tamanho2** o número de colunas.
  - As duas dimensões são, respectivamente, as linhas e as colunas da matriz.

# Matrizes

	col. 1	col. 2	col. 3	...	col. n-1	col. n
linha 1	$x[0][0]$	$x[0][1]$	$x[0][2]$	...	$x[0][n-2]$	$x[0][n-1]$
linha 2	$x[1][0]$	$x[1][1]$	$x[1][2]$	...	$x[1][n-2]$	$x[1][n-1]$
	...	...	...		...	...
linha m	$x[m-1][0]$	$x[m-1][1]$	$x[m-1][2]$	...	$x[m-1][n-2]$	$x[m-1][n-1]$

*$x$  é uma matriz bidimensional  $m \times n$ .*

# Matrizes

- Como definir uma estrutura onde pode-se identificar quatro disciplinas cada uma com 40 alunos?
  - **int disciplinas[4][40];**
  - Comando “for” encadeado para manipular matrizes

```
main( ) {  
    int i, j, matriz[3][3] = { {1, 2, 3}, {4, 5, 6},  
                                {7, 8, 9}};  
    for (i = 0; i < 3; i++) {  
        for(j = 0; j < 3; j++) {  
            printf("%d ", matriz[i][j]);  
        }  
        printf("\n");  
    }  
}
```



# Matrizes

- Assim como vimos com vetores, a inicialização dos valores atribuídos aos elementos de uma matriz também pode ser feita no momento da sua declaração.
- Uma forma alternativa seria preencher a matriz usando loops.



# Exercício

1. Construa um algoritmo que efetue a soma e a impressão do resultado da soma entre duas matrizes  $3 \times 5$ .
2. Faça um programa que multiplica uma matriz  $3 \times 3$  de inteiros por um escalar  $k = 5$ .
3. Dada uma matriz  $5 \times 5$ , elabore um algoritmo que escreva:
  - a) A diagonal principal
  - b) A diagonal secundária
  - c) A soma da linha 4
  - d) A soma da coluna 2