



Universidade Federal do Piauí

Centro de Ensino Aberto e a Distância

Curso de Sistemas de Informação

Algoritmos e Programação I

Vetores e Matrizes

Prof. Arlino Magalhães

arlino@ufpi.edu.br

Sumário

1. Conceito
2. Vetor
3. Matriz

Conceito

Os **vetores e matrizes** são estruturas que permitem armazenar vários dados do mesmo tipo, por isso são chamados de **estruturas homogêneas de dados**.

A utilização destes tipos de estruturas de dados recebe diversos nomes, como: variáveis indexadas, variáveis compostas, variáveis subscritas, arranjos, vetores, matrizes, tabelas em memória ou *arrays*.

Os nomes mais usados e que utilizaremos para estruturas homogêneas são: vetores (matriz de uma linha e várias colunas) e matrizes (vetor de vetores).

Vetor

Sintaxe: *<tipo_de_variável>* *<nome_da_variável>* [*<tamanho_do_vetor>*]

O vetor é um **arranjo de elementos** armazenados em memória principal, um após o outro, onde todos utilizando o mesmo nome de variável.

O vetor possui um conjunto de **índices** que servem para identificar cada um de seus **valores**. O vetor sempre começa com o índice igual 0 (zero).

		Vetor A							
Índices	→	0	1	2	3	4	5	6	7
Valores	→	3,14			0	4	16		-3

Vetor - Exemplo

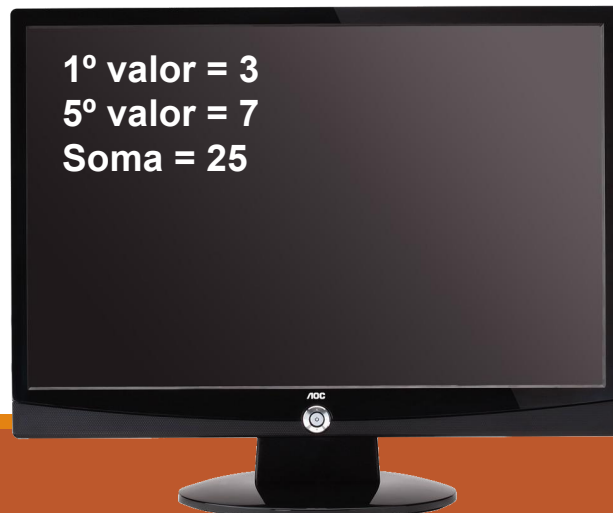
Declaração de uma variável vetor: o número 5 (cinco) indica que o vetor terá cinco elementos.

O vetor sempre começa com índice igual a 0 (zero).

```
#include <stdio.h>
int main() {
    int vetorA [5], soma;
    vetorA [0] = 3;
    vetorA [1] = 4;
    vetorA [2] = 0;
    vetorA [3] = 11;
    vetorA [4] = 7;
    soma = vetorA [0] + vetorA [1] +
           vetorA [2] + vetorA [3] + vetorA [4];
    printf ("1º valor = %d \n", vetorA [0]);
    printf ("5º valor = %d \n", vetorA [4]);
    printf ("Soma = %d \n", soma);
    return 0;
}
```

Vetor - Executando o exemplo

soma		vetorA
25	0	3
	1	4
	2	0
	3	11
	4	7



```
#include <stdio.h>
int main() {
    int vetorA [5], soma;
    vetorA [0] = 3;
    vetorA [1] = 4;
    vetorA [2] = 0;
    vetorA [3] = 11;
    vetorA [4] = 7;
    soma = vetorA [0] + vetorA [1] +
           vetorA [2] + vetorA [3] + vetorA [4];
    printf ("1º valor = %d \n", vetorA [0]);
    printf ("5º valor = %d \n", vetorA [4]);
    printf ("Soma = %d \n", soma);
    return 0;
}
```

Vetor - Exemplo

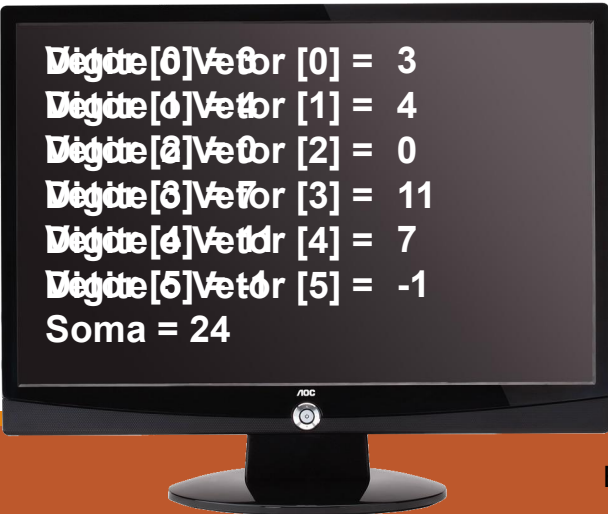
Os *loops* podem ser utilizados para percorrer um vetor.

```
#include <stdio.h>
int main(){
    int i, soma, vet [6000];
    for (i = 0; i < 6000; i++) {
        printf ("Digite o Vetor [%d]:", i);
        scanf ("%d", &vet [i]);
    }
    for (i = 0; i < 6000; i++)
        printf ("Vetor [%d] = %d \n", i, vet [i]);
    soma = 0;
    for (i = 0; i < 6000; i++)
        soma += vet [i];
    printf ("Soma = %d \n", soma);
    return 0;
}
```

Vetor - Exemplo

i	soma
0	0
1	3
2	7
3	7
4	18
5	25
6	24

	Vet
0	3
1	4
2	0
3	11
4	7
5	-1



Digite o Vetor [0] = 3
Digite o Vetor [1] = 4
Digite o Vetor [2] = 0
Digite o Vetor [3] = 11
Digite o Vetor [4] = 7
Digite o Vetor [5] = -1
Soma = 24

```
#include <stdio.h>
int main() {
    int i, soma, vet [6];

    for (i = 0; i < 6; i++) {
        printf ("Digite o Vetor [%d]:", i);
        scanf ("%d", &vet [i]);
    }

    for (i = 0; i < 6; i++)
        printf ("Vetor [%d] = %d \n", i, vet [i]);

    soma = 0;
    for (i = 0; i < 6; i++)
        soma += vet [i];

    printf ("Soma = %d \n", soma);

    return 0;
}
```


Exercício 01

Crie um algoritmo que dados dois vetores, seja gerado um terceiro vetor onde cada um de seus elementos é a soma dos respectivos elementos dos dois primeiros vetores.

Exemplo:

A		B		C
5		3		8
0		4		4
-1	+	0	=	-1
3		1		4
3		2		5

```
#include <stdio.h>
int main() {
    int a [50], b [50], c [50];
    for (int i = 0; i < 50; i++) {
        printf ("Digite A [%d]: ", i);
        scanf ("%d", &a [i]);
        printf ("Digite B [%d]: ", i);
        scanf ("%d", &b [i]);
    }
    for (int i = 0; i < 50; i++)
        c [i] = a [i] + b [i];
    return 0;
}
```

Exercício 02

Dado um vetor de 100 números inteiros, informar se cada número do vetor é par ou impar.

```
#include <stdio.h>
int main() {
    int vetor [100];
    for (i = 0; i < 100; i++) {
        printf ("Digite o %dº numero: ", i);
        scanf ("%d", &vetor [i]);
    }
    for (i = 0; i < 100; i++)
        if ( (vetor [i] % 2) == 0)
            printf ("O numero %d eh par!", vetor [i]);
        else
            printf ("O numero %d eh impar!", vetor [i]);
    return 0;
}
```

Exercício 03

Dado um vetor de 300 números inteiros, informar qual o maior número.

```
#include <stdio.h>
int main(){
    int vetor [300], maior;
    for (i = 0; i < 300; i++){
        printf ("Digite o %dº numero: ", i);
        scanf ("%d", &vetor [i]);
    }
    maior = vetor [0];
    for (i = 1; i < 300; i++)
        if (vetor [i] > maior)
            maior = vetor [i];
    printf ("Maior =%d \n", maior);
    return 0;
}
```

Exercício 04

Dada uma *string*, exibir os seus caracteres um a um.

Exemplo:

string							
0	1	2	3	4	5	6	7
J	o	ã	o	\0			

Uma *string* é um vetor de caracteres.

O caractere invisível '\0' marca o fim de uma *string*.

```
#include <stdio.h>
#include <string.h>
int main() {
    char string [100], i;
    printf ("Digite um nome:");
    gets (string);

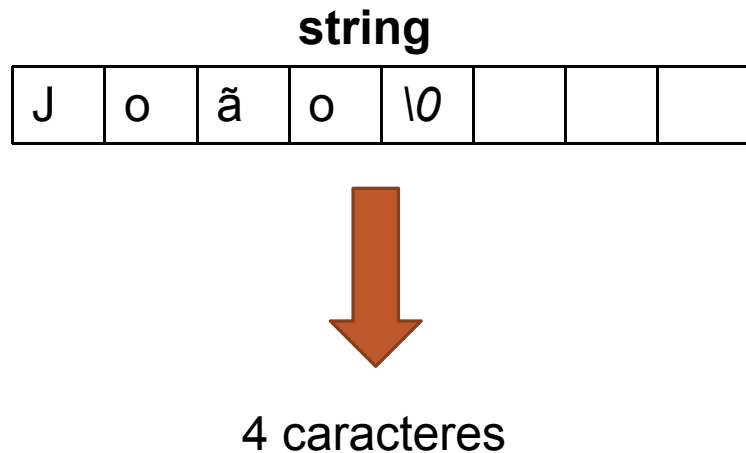
    for (i = 0; string [i] != '\0'; i++)
        printf ("Caractere %d = %c \n", i, string [i]);

    return 0;
}
```

Exercício 05

Dada uma *string*, mostrar quantos caracteres ela possui.

Exemplo:



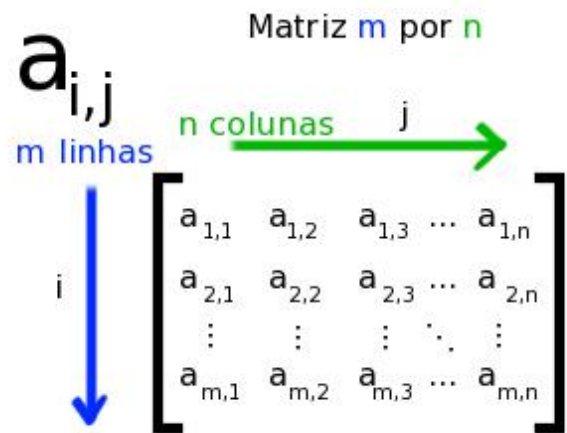
```
#include <stdio.h>
#include <string.h>
int main(){
    char nome[100], quant = 0;

    printf ("Digite um nome:");
    gets (nome);
    for (i = 0; nome [i] != '\0'; i++)
        quant++;
    printf ("O nome %s possui %d
           caracteres! ", nome, quant);

    return 0;
}
```

Matriz

Uma matriz é um vetor com mais de uma dimensão.



Índices

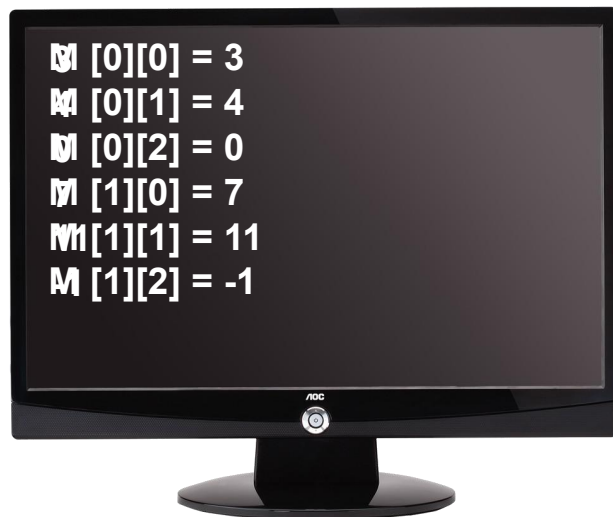
Matriz M

	0	1	2	3
0				
1			17	
2	3,14			
3				
4				

Matriz - Exemplo

i	j
0	0
1	1
2	2
	3

	0	1	2
0	3	4	0
1	7	11	-1



```
#include <stdio.h>
int main(){
    int i, j, mat [2][3];

    for (i = 0; i < 2; i++)
        for (j = 0; j < 3; j++)
            scanf ("%d", mat [i][j]);

    for (i = 0; i < 2; i++)
        for (j = 0; j < 3 ; j++)
            printf ("M [%d][%d] = %d: ",
                    i, j, mat [i][j]);

    return 0;
}
```

Exercício 06

Dadas duas matrizes A e B, gerar a matriz soma C.

A			B		
1	1	1	5	7	2
3	4	3	0	2	3
3	0	9	3	0	0

+

=


↓

C		
6	8	3
3	6	6
6	0	9

```
#include <stdio.h>
int main(){
    int i, j, a [3][3], b [3][3], c [3][3];
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++) {
            printf ("Digite a [%d][%d]:", i, j);
            scanf ("%d", a [i][j]);
            printf ("Digite b [%d][%d]:", i, j);
            scanf ("%d", b [i][j]);
        }
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3 ; j++)
            c [i][j] = a [i][j] + b [i][j];
    return 0;
}
```


Exercício 07

Dada uma matriz A 10 x 20, construir uma matriz B cujos elementos são o triplo de cada elemento de A.

A				B		
1	1	1		3	3	3
3	4	3		9	12	9
3	0	9		9	0	27

```
#include <stdio.h>
int main() {
    int i, j, a [10][30], b [10][30];

    for (i = 0; i < 10; i++)
        for (j = 0; j < 30; j++) {
            printf ("Digite a [%d][%d]:", i, j);
            scanf ("%d", a [i][j]);
        }
    for (i = 0; i < 10; i++)
        for (j = 0; j < 30; j++)
            b [i][j] = 3*a [i][j];

    return 0;
}
```

Exercício 08

Dada uma matriz A numérica, exibir o somatório de seus números.

A

1	1	1
3	4	3
3	0	9



Somatório = 25

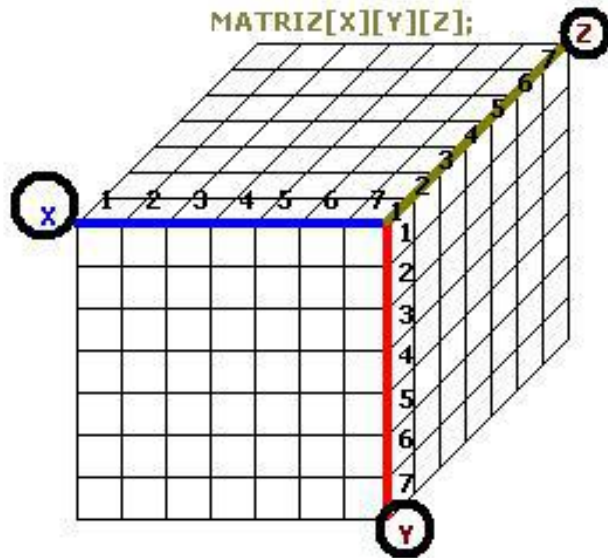
```
#include <stdio.h>
int main() {
    int i, j, a [10][30], soma = 0;

    for (i = 0; i < 10; i++)
        for (j = 0; j < 30; j++) {
            printf ("Digite a [%d][%d]:", i, j);
            scanf ("%d", a [i][j]);
        }

    for (i = 0; i < 10; i++)
        for (j = 0; j < 30; j++)
            soma += a [i][j];
    printf ("Somatório = %d", soma);
    return 0;
}
```

Exercício 09

Leia uma matriz numérica de três dimensões e liste seus valores em seguida.



```
#include <stdio.h>

int main() {
    int x, y, z, m [10][10][10];
    for (x = 0; x < 10; x++)
        for (y = 0; y < 10; y++)
            for (z = 0; z < 10; z++) {
                printf ("Digite m [%d][%d]
                        [%d]:", x, y, z);
                scanf ("%d", m [x][y][z]);
            }
    for (x = 0; x < 10; x++)
        for (y = 0; y < 10; y++)
            for (z = 0; z < 10; z++)
                printf ("m[%d][%d][%d]: %d",
                        x, y, z, a [x][y][z]);

    return 0;
}
```