

# Laboratório de Programação

**Prof. Dr. Paulo Rogério de Almeida Ribeiro**

Coordenação do Curso de Engenharia da Computação

Vetores e matrizes



# Arranjos

# Arranjos

- Coleção de variáveis do mesmo tipo;

## Arranjos

- Coleção de variáveis do mesmo tipo;
- Referenciado por um nome em comum;

## Arranjos

- Coleção de variáveis do mesmo tipo;
- Referenciado por um nome em comum;
- Elemento específico acessado via índice;

# Arranjos

- Coleção de variáveis do mesmo tipo;
- Referenciado por um nome em comum;
- Elemento específico acessado via índice;
- Exemplos: vetor, matriz e string;

# Arranjos

- Coleção de variáveis do mesmo tipo;
- Referenciado por um nome em comum;
- Elemento específico acessado via índice;
- Exemplos: vetor, matriz e string;
- Arranjos unidimensionais: vetores e strings;

# Arranjos

- Coleção de variáveis do mesmo tipo;
- Referenciado por um nome em comum;
- Elemento específico acessado via índice;
- Exemplos: vetor, matriz e string;
- Arranjos unidimensionais: vetores e strings;
- Arranjos multidimensionais: matriz.



## Arranjo

- Sintaxe:
- *Tipo nome [numero\_de\_elementos] = { INICIALIZAÇÃO};*
  - INICIALIZAÇÃO é opcional.

## Arranjo

- Sintaxe:
- *Tipo nome [numero\_de\_elementos] = { INICIALIZAÇÃO};*
  - **INICIALIZAÇÃO** é opcional.

- Exemplos:

- `int vet[10];`
- `char strings[20];`
- `float novo [5] = {1,2,3,4,5};` //com inicialização
- `int vetor []={4,4,4};` //com inicialização
- `int vec[5] = {0};` //inicializa todos os elementos com 0

## Acesso ao conteúdo

## Acesso ao conteúdo

- Através dos índices;

## Acesso ao conteúdo

- Através dos índices;
- Índice começa de 0 e vai até TAMANHO-1;

## Acesso ao conteúdo

- Através dos índices;
- Índice começa de 0 e vai até TAMANHO-1;
  - vetor[0] - primeiro elemento;

## Acesso ao conteúdo

- Através dos índices;
- Índice começa de 0 e vai até TAMANHO-1;
  - vetor[0] - primeiro elemento;
  - vetor[1] - segundo elemento;

## Acesso ao conteúdo

- Através dos índices;
- Índice começa de 0 e vai até TAMANHO-1;
  - vetor[0] - primeiro elemento;
  - vetor[1] - segundo elemento;
  - vetor[TAMANHO-1] - último elemento;



## Exemplo de acesso ao conteúdo

- **int vetor [3] = {1,2,3}**

1	2	3
[0]	[1]	[2]

- **vetor [0] = 1**

- **vetor [1] = 2**

- **vetor [2] = 3**

## Exemplo de preenchimento do vetor - laço

```
■ for(int i=0; i<5; i++)  
    vetor[i] = 10;
```

- Exemplo 1- Imprimindo um vetor de tamanho 50:

```
#include <stdio.h>
int main(){
int vec [50];
int i;
for (i=0;i<50;i++){
    vec[i]=i;
    printf("%d", vec[i]);
}
return 0;
}
```

## Exemplo: inicializa um vetor de 5 posições ( via teclado)

```
#include<stdio.h>
int main (){
int vetor[5] , i;
for ( i=0; i<5; i++ ) {
    printf ("\nElemento[%d]= ", i);
    scanf ("%d", &vetor[ i ]);
}
printf ("\n*****Imprimindo*****\n");
for (i=0;i<5;i++){
    printf("%d", vetor[i]);
}
return 0;
}
```


## Exercícios

- 1) Crie um programa que leia 100 valores e mostre-os na ordem inversa da leitura.

## Exercícios

- 1) Crie um programa que leia 100 valores e mostre-os na ordem inversa da leitura.
- 2) Dado um vetor e um número informe se este número está no vetor.

## String - vetor de caracteres

- Exemplos:
- `char teste[4]="ola";`
- `char teste[]="ola";`
- `char teste []={'o','l','a'};` 

## String - vetor de caracteres

- Função `printf()`
  - `printf("Nome: %s",vetorNome);`



## String - vetor de caracteres

- Função `printf()`
  - `printf("Nome: %s",vetorNome);`
- Função `puts()`
  - `puts(vetorNome);` //já inclui o `\n`

## String - vetor de caracteres

- Função `printf()`
  - `printf("Nome: %s", vetorNome);`
- Função `puts()`
  - `puts(vetorNome);` //já inclui o `\n`
- Função `scanf()`
  - `scanf("%s", nome_string);` //OBS: não precisa do `&`

## Exemplo de printf e scanf

```
#include<stdio.h>
int main(void) {
    char nome[50];
    printf(" Digite seu nome: " );
    scanf(" %s",nome);
    printf(" O nome armazenado foi: %s" , nome);
    return 0;
}
```

gets - similar ao scanf mas inclui espaço

- Exemplo gets():

```
#include<stdio.h>
```

```
int main(void) {
```

```
    char nome[50];
```

```
    printf("Digite seu nome: ");
```

```
    gets(nome);
```

```
    printf("O nome armazenado foi: %s", nome);
```

```
    return 0;
```

```
}
```

## Biblioteca String

### ▪ Header: **<string.h>**

Nome	Função
strcpy(s1, s2)	Copia s2 em s1.
strcat(s1, s2)	Concatena s2 ao final de s1.
strlen(s1)	Retorna o tamanho de s1.
strcmp(s1, s2)	Retorna 0 se s1 e s2 são iguais; menor que 0 se s1<s2; maior que 0 se s1>s2.
strchr(s1, ch)	Retorna um ponteiro para a primeira ocorrência de ch em s1.
strstr(s1, s2)	Retorna um ponteiro para a primeira ocorrência de s2 em s1.

## Exemplo biblioteca String

```
#include<stdio.h>
#include<string.h>
int main () {
    char nome1[80],nome2[80];
    printf("Digite o primeiro nome:");
    scanf("%s",nome1);
    printf("Digite o segundo nome:");
    scanf("%s",nome2);
    printf("O tamanho da string %s é %d\n", nome1,strlen(nome1));
    printf("O tamanho da string %s é %d", nome2,strlen(nome2));
    return 0;
}
```

## Exercícios

- 1) Crie um programa que leia uma string e informe se essa tem a vogal 'a'.

## Exercícios

- 1) Crie um programa que leia uma string e informe se essa tem a vogal 'a'.
- 2) Crie um programa que leia uma string e informa quantas vogais essa tem.



# Matriz

Vetor multidimensional

## Matriz

Vetor multidimensional

- Sintaxe:

- *Tipo* *nome* [*tamanho1*][*tamanho2*]... [*tamanhoN*] = {  
INICIALIZAÇÃO};

- Exemplos:

- int vetor [3][2];
- char strings[3][3];
- float novo [2][2] = {1,2,3,4};      //com inicialização

## Índices da matriz

- Índices começam em 0 e vão até (TAMANHO – 1):
  - `matriz[0][0]` - primeiro elemento
  - `matriz[tamanho1-1][tamanho2-1]` - último elemento

## Matriz

num [t] [i]					
		0	1	2	3
0	1	2	3	4	
1	5	6	7	8	
2	9	10	11	12	

## Exemplo de matriz

Inicialize e imprima uma matriz 2x2

```
#include<stdio.h>
int main (){
int matriz[2][2] , linha, coluna;
printf (" Digite os elementos da matriz:\n" );
for ( linha=0; linha<2; linha++ )
    for ( coluna=0; coluna<2; coluna++ ) {
        printf ("\nElemento[%d][%d] = ", linha, coluna);
        scanf (" %d", &matriz[ linha ][ coluna ]);
    }
printf("\n***** Imprimindo ***** \n");
for ( linha=0; linha<2; linha++ )
    for ( coluna=0; coluna<2; coluna++ ) {
        printf (" Elemento[%d][%d] = %d\n", linha, coluna,matriz[ linha ][ coluna ]);
    }
return 0;
}
```

## Exercícios

1) Escreva um programa que receba uma matriz 2x2 e informa o maior elemento da matriz, assim como sua respectiva posição, ou seja, linha e coluna que o elemento ocupa.

## Exercícios

- 1) Escreva um programa que receba uma matriz  $2 \times 2$  e informe o maior elemento da matriz, assim como sua respectiva posição, ou seja, linha e coluna que o elemento ocupa.
- 2) Crie um programa que leia 2 matrizes  $2 \times 2$  e some-as, bem como mostra a matriz resultante.

## Definir tamanho antecipadamente

- Palavra reservada **const**:
  - **const nome\_da\_constante = valor;**
- Exemplo:  
**const int** num=10;  
**int** vetor[num];



## Definir tamanho antecipadamente

- Exemplo:

```
#define TAM 5
```

```
int vetor[TAM];
```

```
for(i=0; i<TAM; i++)  
    vetor[i] = TAM - i;
```