

## Aula 5

### Lógica de Programação e Algoritmos

Prof. Sandro de Araujo

### Conversa Inicial

- Esta aula apresenta a seguinte estrutura de conteúdo:
  - Vetor e matriz
  - Declarar e inicializar vetores
  - Declarar e inicializar matrizes
  - Cadeia de caracteres
  - Inicializar uma cadeia de caracteres

- O objetivo dessa aula é conhecer os principais conceitos e aplicações dos vetores, matrizes e cadeias de caracteres – e de que forma declará-los e inicializá-los – bem como representá-los em pseudocódigo e linguagem C para resolver problemas computacionais

### Vetor e matriz

#### Vetor ou *array*

- Variável composta homogênea unidimensional
  - Conjunto de dados do mesmo tipo
- Possuem o mesmo identificador (nome)
- Acessados por um número inteiro chamado de *índice*
- Tipo\_de\_dado + Nome\_do\_Vetor + [Posições - Índices]

Índice: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]  
 Nome: temp 

18	17	20	26	32	29	15	12	16	21
----	----	----	----	----	----	----	----	----	----

Fonte: Puga e Rissetti, 2016.

## Matriz

- É um vetor de vetores
- Acessado também por índice
- Estrutura de dados com duas ou mais dimensões
- Estrutura composta por linhas e colunas
- Tipo\_de\_dado + Nome\_da\_Matriz + [Qtde Linhas] + [Qtde Colunas]

	COLUNAS	
LINHAS	0,0	0,1
	1,0	1,1

19	21
33	581

Índices	Dados
[0,0]	19
[0,1]	21
[1,0]	33
[1,1]	581


## Declarar e inicializar vetores

- A declaração de um vetor deve conter três informações:
  - Nome do vetor (identificador)
  - Número de posições (tamanho)
  - Tipo de dado que será armazenado

### Sintaxe pseudocódigo e linguagem C

- **Pseudocódigo**
  - nomeDoVetor: vetor [início..final] de tipoDoVetor
  - nota: vetor[1..4] de Inteiro
- **Linguagem C**
  - tipoDoVetor nomeDoVetor [QuantidadeDeElementos]
  - int nota [4]

### Declarar e inicializar vetores

- 
- ```
int números[6] = {1,2,3};
```
- ① 

```
int números[6] = {1,2,3,0,0,0};
```
- Também é possível inicializar um vetor sem especificar a quantidade de elementos:  

```
int números[ ] = {3,6,7};
```
  - Ou seja: 

```
números[0] = 3;
```

```
números[1] = 6;
```

```
números[2] = 7;
```

### Pseudocódigo

Exibindo os Valores das Notas

```
NOTA 1: 7.6
NOTA 2: 8.9
NOTA 3: 9
NOTA 4: 9.9

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

### Linguagem C

....

```
"C:\Users\Casa\Documents\Sandro\FACULDADES\UNINTER\Linguagem de Programação\teste1\bin\Debug\teste1.exe"
Exibindo os Valores das Notas

NOTA 1: 7.6
NOTA 2: 8.9
NOTA 3: 9.0
NOTA 4: 9.9
Pressione qualquer tecla para continuar. . .

....
```

### Declarar e inicializar matrizes

- A declaração de uma matriz deve conter quatro informações:
  1. Nome da matriz (identificador)
  2. Número de linhas (tamanho)
  3. Número de colunas (tamanho)
  4. Tipo de dado que será armazenado

### Sintaxe pseudocódigo e linguagem C

- Pseudocódigo
  - nomeDaMatriz: vetor[início..final, início..final] de tipoDoVetor
  - nota: vetor[1..4,1..4] de Inteiro
- Linguagem C
  - tipoDaMatriz nomeDaMatriz [QtdeLinhas][QtdeColunas];
  - int nota [4][4];

### Declarar e inicializar vetores

- Para declarar uma matriz 2X4 e inicializá-la, devemos colocar cada linha entre chaves {}, e separar elas por vírgulas:

#### Exemplo:

- int numeros[2][4] = { {11, 22, 35, 54 }, {5, 9, 16, 8 } };

|    |    |    |    |
|----|----|----|----|
| 11 | 22 | 35 | 54 |
| 5  | 9  | 16 | 8  |

Quatro colunas

Duas linhas

### Pseudocódigo

- ...
- var
- tabela: vetor[1..3,1..2] de inteiro
- linha,coluna: inteiro
- (...)

(...)

#### início

- tabela[0,0] <- 1
- tabela[0,1] <- 8
- tabela[1,0] <- 3
- tabela[1,1] <- 9
- tabela[2,0] <- 6
- tabela[2,1] <- 78955

- escreval ("Imprimindo a matriz tabela:")
- escreval (" ")
- para linha de 0 até 2 faça
- para coluna de 0 até 1 faça
- escreval("tabela[",linha,"][",coluna,"]=",tabela[linha,coluna])
- fimpara
- fimpara

```
Imprimindo a matriz tabela:
tabela[ 0][ 0] 1
tabela[ 0][ 1] 8
tabela[ 1][ 0] 3
tabela[ 1][ 1] 9
tabela[ 2][ 0] 6
tabela[ 2][ 1] 78955
*****
O conteúdo da tabela[2][1] é: 78955
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

## Linguagem C (continuação)

```
int main()
{
    int tabela[3][2] = {{1,8},{3,9},{6,78955}};
    int linha,coluna;
```

```
printf("Imprimindo a matriz tabela: \n\n");
for(linha=0;linha<3;linha++)
{
    for(coluna=0;coluna<2;coluna++)
    {
        printf("tabela[%d][%d]=%d",linha,coluna,tabela[linha][coluna]);
        printf("\n");
    }
}
```

## Linguagem C

```
Imprimindo a matriz tabela:
tabela[0][0]=1
tabela[0][1]=8
tabela[1][0]=3
tabela[1][1]=9
tabela[2][0]=6
tabela[2][1]=78955
*****
O conteudo da tabela[2][1] e: 78955
Pressione qualquer tecla para continuar. . .
```

## Pseudocódigo

## Linguagem C

| Pseudocódigo                | Linguagem C                 |
|-----------------------------|-----------------------------|
| Imprimindo a matriz tabela: | Imprimindo a matriz tabela: |
| tabela[ 0][ 0] = 1          | tabela[0][0]=1              |
| tabela[ 0][ 1] = 8          | tabela[0][1]=8              |
| tabela[ 1][ 0] = 3          | tabela[1][0]=3              |
| tabela[ 1][ 1] = 9          | tabela[1][1]=9              |
| tabela[ 2][ 0] = 6          | tabela[2][0]=6              |
| tabela[ 2][ 1] = 78955      | tabela[2][1]=78955          |
| *****                       | *****                       |

## Cadeias de caracteres

## Cadeia de caracteres (*string*)

É uma variável usada para armazenar cadeia de caracteres com conteúdo que pode ser alterado ou substituído por outros elementos para formar uma nova cadeia de caracteres

- A linguagem C não possui um tipo de dado similar à *string*. Em vez disso, para armazenar uma cadeia de caracteres, utiliza vetores (matrizes unidimensionais)
  - `char NomeDaString[NúmeroDeElementos]`
  - `char nome[8];`

- A palavra "UNINTER", na linguagem de programação C, é um *array* que contém oito [8] posições, conforme mostrado abaixo:
  - `char nome [8] = { 'U', 'N', 'I', 'N', 'T', 'E', 'R', '\0' };`

### Inicializar uma cadeia de caracteres

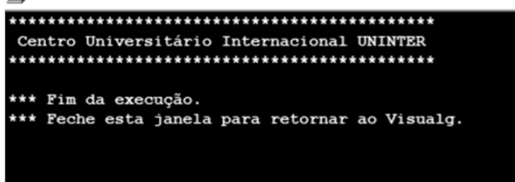
- Uma *string* pode ser inicializada automaticamente pelo programa ou pode receber um valor por meio do teclado
- Usam-se as duas formas para inicializar uma *string* em C:
  - `char nome [ ] = "UNINTER";`
  - `char nome [ ] = { 'U', 'N', 'I', 'N', 'T', 'E', 'R', '\0' };`

- Exemplo 1
  - `char nome [ ] = "UNINTER";`
  - `char nome [ ] = { 'U', 'N', 'I', 'N', 'T', 'E', 'R', '\0' };`
- Exemplo 2
  - `char [ ] = "12345";`
  - `char [ ] = { '1', '2', '3', '4', '5', '\0' };`
  - ✓ Não é possível fazer cálculos matemáticos com números do tipo *caractere*

### Pseudocódigo

- algoritmo "exemploString"
- var
  - nome1 : caractere
  - nome2 : caractere
  - nome3 : caractere
  - asteriscos : caractere

- (...)
  - inicio
    - nome1 <- "Centro Universitário"
    - nome2 <- "Internacional"
    - nome3 <- "UNINTER"
    - asteriscos <- "\*\*\*\*\*"
    - escreval(asteriscos)
    - escreval(" ",nome1," ",nome2," ",nome3)
    - escreval(asteriscos)
  - finalgoritmo

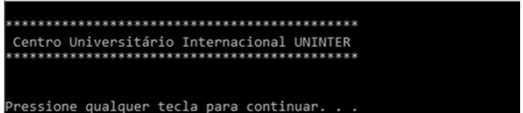


```
*****
Centro Universitário Internacional UNINTER
*****
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

### Linguagem C

- #include<stdio.h>
- #include<conio.h>
- #include <locale.h>
- int main()
- {
- char nome1[ ]= "Centro Universitário";
- char nome2[ ]= "Internacional";
- char nome3[ ]= "UNINTER";
- char asteriscos[]= "\*\*\*\*\*\n";

- (...)
  - printf("%s",asteriscos);
  - printf("%s %s %s", nome1,nome2,nome3, setlocale(LC\_ALL,""));
  - printf("%s\n\n",asteriscos);
  - system("pause");
  - return 0;
  - }



```
*****
Centro Universitário Internacional UNINTER
*****
Pressione qualquer tecla para continuar. . .
```

### Pseudocódigo



```
*****
Centro Universitário Internacional UNINTER
*****

Linguagem C

*****
Centro Universitário Internacional UNINTER
*****
```

