INTRODUÇÃO À PROGRAMAÇÃO

Aula 12 - Vetores e Variáveis Textuais

Prof. Carlos Alexandre Siqueira da Silva

Campus de Alegre

Revisão da Aula Anterior

- Introdução ao conceito de Vetores
  - Vantagens
  - Uso de índices
- Operações com Vetores
  - Declaração
  - Inicialização
  - Atribuição
- Exemplos

Variáveis textuais (cadeias de caracteres)

- Variáveis textuais são vetores de caracteres.
- Também chamadas de **strings** em C.
- Armazenam letras, números ou caracteres especiais ($, @, !, &, etc).

Sintaxe:

```
char Nome[20];
char Curso[10] = {'T', 'A', 'D', 'S'};
```

Comando de Entrada - Variáveis Textuais

- Declaração da variável textual:
  ```
  char Nome[30];
  ```

- Leitura de uma cadeia de caracteres:
  ```
  scanf("%s", Nome);
  ```

- Leitura de caracteres separadamente:
  ```
  scanf("%c", Nome[0]);
  scanf("%c", Nome[1]);
  ```

Comando de Saída - Variáveis Textuais

- Declaração da variável textual:
  ```
  char Nome[30];
  ```

- Escrita de uma cadeia de caracteres:
  ```
  printf("O nome é %s\n", Nome);
  ```

- Escrita de caracteres separadamente:
  ```
  scanf("A primeira letra é %c\n", Nome[0]);
  scanf("A segunda letra é %c\n", Nome[1]);
  ```

Manipulação de Variáveis Textuais - Exemplo 1

Exemplo 1 – Mostrar uma String na tela, um caracter por vez

```
1 char Texto[15] = {'T', 'e', 's', 't', 'a', 'n', 'd', 'o'};
2 int Cont;
3 printf("%c", Texto[0]);
4 for(Cont=1; Cont<15; Cont++){
5       printf(", %c", Texto[Cont]);
6 }
```

Resultado esperado:

T, e, s, t, a, n, d, o, , , , , , , ← (observe os espaços em branco no final)

Terminador Nulo em Variáveis Textuais

- Toda string termina com um **Caracter Nulo** '\0'.
- Indica o fim do texto.

Manipulação de Variáveis Textuais - Exemplo 2

Exemplo 2 – Mostrar uma String na tela, um caracter por vez

```
1 char Texto[15] = {'T', 'e', 's', 't', 'a', 'n', 'd', 'o', '\0'};
2 int Cont = 1;
3 printf("%c", Texto[0]);
4 while(Texto[Cont] != '\0'){
5     printf(", %c", Texto[Cont]);
6     Cont++;
7 }
```

Resultado esperado:

T, e, s, t, a, n, d, o ← (sem os espaços em branco)

Funções para Variáveis Textuais (1/5)

Como String são vetores de caracteres, não é possível compará-los ou atribuí-los da mesma maneira que variáveis primitivas.

- Trecho válido:

```
int Valor1, Valor2;
if (Valor1 == Valor2){}
Valor1 = 10;
```

- Trecho incorreto:

```
char Texto1[10], Texto2[10];
if (Texto1 == Texto2){} ← Não funciona!
Texto1 = "Teste"; ← Não funciona!
```

Funções para Variáveis Textuais (2/5)

Função **strlen();**
Retorna a quantidade de caracteres de uma string.

Sintaxe:

strlen(<Variável Textual>);

Exemplo:

```
1 char nome[10] = "Carlos";
2 printf("Tamanho: %d\n", strlen(nome)); //Saida: 6
```

Funções para Variáveis Textuais (3/5)

Função **strcpy();**
Realiza a cópia de uma string para outra.

Sintaxe:

strcpy(<Variável de destino>, <Variável de Origem>);

Exemplo:

```
1 char destino[20];
2 strcpy(destino, "Teste");
3 printf("%s\n", destino);    // Saida: Teste
```

Funções para Variáveis Textuais (4/5)

Função **strcat();**
Concatena (anexa) uma string ao final da outra.

Sintaxe:

strcat(<Variável de destino>, <Variável de Origem>);

Exemplo:

```
1 char texto[30] = "Ola ";
2 strcat(texto, "pessoal!");
3 printf("%s\n", texto);    // Saida: Ola pessoal!
```

Funções para Variáveis Textuais (5/5)

Função **strcmp();**

Compara duas strings S1 e S2. Retorna:

- 0, se as duas forem iguais.
- Valor negativo, se S1 < S2.
- Valor positivo, se S1 > S2.

Sintaxe:

strcmp(<Variável 1>, <Variável 2>);

Exemplo:

```
1 printf("%d\n", strcmp("abc", "abc"));  // Saida: 0
2 printf("%d\n", strcmp("abc", "abd"));  // Saida: Valor negativo
```