



String - Programação II

Autor Prof.: Delcino Picinin Júnior

String

O termo string serve para identificar uma sequência de caracteres.

Na prática, as strings são usadas para representar textos.

Em linguagem C, ao contrário de outras linguagens, não existe um tipo de dados string nativo.

Para representar uma string em C, devemos criar um vetor de caracteres, ou seja um vetor do tipo char.

Código 1: Exemplo de Código em C

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void main(){
5
6 char a [20];
7 int x,y;
8
9 scanf("%[^\\n]s",a); // scanf que permite ler espaços em branco
10 scanf("%*c"); //limpa o buffer do teclado do scanf %[^\n]s
11
12 printf ("Digitou %s ",a);
13
14 x=strlen(a);
15 printf ("\n Tamanho %d",x);
16
17 for (y=0; y<=x; y++){
18 printf ("\n %c %d",a[y],a[y]);
19 }
20
21 }
```

strcpy

Sintaxe: strcpy(string_destino, string_origem);

Realiza a cópia do conteúdo de uma variável a outra.

Obs: Ambas devem ser strings.

Código 2: Exemplo de Código STRCPY

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strcpy
3
4 void main (){
5     char nome[15];
6
7     strcpy(nome, "Fulano de Tal");
8     //strcpy(string_destino, string_origem);
9     //note que a string de destino é nome
10    //a string de origem é "Fulano de Tal"
11
12    printf("Nome = %s", nome);
13
14 }
```

strncpy

Sintaxe: strncpy(string_destino, string_origem, tamanho);

Realiza a cópia do conteúdo de uma variável a outra, porém, deve ser especificado o tamanho a ser copiado.

Obs: Tanto a variável de destino como a de origem devem ser strings.

Código 3: Exemplo de Código STRNCPY

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strncpy
3
4 void main (){
5     char str1[8] = "Curso C";
6     char str2[5];
7
8     strncpy(str2, str1, 5);
9
10    str2[5] = '\0';
11    printf("str2 = %s\n", str2);
12
13 }
```

strcat

Sintaxe: strcat(string_destino, string_origem);

Realiza a concatenação do conteúdo de uma variável a outra. Ambas devem ser strings.

Código 4: Exemplo de Código STRCAT

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strcat
3
4 void main (){
5     char str [10] = "Curso";
6     strcat ( str , " de C");
7     //Concatena a string " de C" com o conteúdo da string str
8
9     printf ("str = %s\n", str);
10    //Será exibido curso de C
11
12 }
```

strncat

Sintaxe: strncat(string_destino, string_origem, tamanho);

Realiza a concatenação do conteúdo de uma variável a outra, porém, deve ser especificado o tamanho a ser concatenado. Ambas devem ser strings.

Código 5: Exemplo de Código STRNCAT

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strncat
3
4 void main (){
5     char str1 [20] = "Curso";
6     char str2 [17] = " de programacao C";
7
8     strncat (str1 , str2 , 15);
9     //concatena a string1 com 15 posições da string2
10
11     printf ("str1 = %s\n", str1);
12     //Será exibido Curso de Programação
13
14 }
```

strlen

Sintaxe: variável do tipo inteiro = strlen(string);
Determina o tamanho de uma string.

Código 6: Exemplo de Código STRLEN

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strlen
3
4 void main (){
5     char str [5] = "Curso";
6     int tamanho;
7
8     tamanho = strlen(str);
9
10    printf ("O tamanho da string %s vale %d\n", str, tamanho);
11
12 }
```

strcmp

Sintaxe: variável do tipo inteiro = strcmp(string1, string2);
Compara o conteúdo de duas strings;
Possíveis valores de retorno:

- 0: conteúdo das strings são iguais
- < 0: conteúdo da string1 é menor do que string2
- > 0: conteúdo da string1 é maior do que string2

Código 7: Exemplo de Código STRCMP

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strcmp
3
4 void main (){
5     char str1 [4] = "abc";
6     char str2 [4] = "abd";
7     int retorno;
8
9     retorno = strcmp(str1, str2);
10    printf ("retorno = %d\n", retorno);
11    //mostra o retorno da função strcmp
12
13 }
```

strncmp

Sintaxe: variável do tipo inteiro = strncmp(string1, string2, tamanho);

Também faz a comparação do conteúdo de duas strings, porém, deve ser especificado o tamanho a ser comparado;

Possíveis valores de retorno:

- 0: conteúdo das strings são iguais
- < 0: conteúdo da string1 é menor do que string2
- > 0: conteúdo da string1 é maior do que string2

Código 8: Exemplo de Código STRNCMP

```
1 #include <stdio.h>
2 #include <string.h> //necessário para strncmp;
3
4 void main (){
5     char str1 [15] = "Curso de C";
6     char str2 [15] = "Curso de Java";
7     int retorno;
8
9     retorno = strncmp(str1, str2, 5);
10
11     printf ("retorno = %d\n", retorno);
12
13 }
```
