

Algoritmos e Programação II

Funções de String

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

O que é uma String?

String é uma seqüência de caracteres utilizada para o armazenamento de texto.

Na linguagem C strings são vetores de caracteres que possuem um *character* que indica o término de seu conteúdo, o *character* nulo ‘\0’ (contrabarra zero).

BIBLIOTECA STRING.H

- A biblioteca string da linguagem C contém várias funções de manipulação de strings. Para usar essas funções, o seu programa deve incluir a interface [string.h](#):

```
#include <string.h>
```

BIBLIOTECA STRING.H

A biblioteca **string.h** da linguagem C, contém uma série de funções para manipular strings:

- Copiar strings em C usando `strcpy` e `strncpy`;
- Concatenar strings em linguagem C usando `strcat` e `strncat`;
- Descobrir o tamanho de uma string em C usando `strlen()`;
- Comparar strings em C usando `strcmp()`;

STRCPY

Sintaxe:

```
strcpy(string_destino, string_origem);
```

- Realiza a cópia do conteúdo de uma variável a outra.
- Obs: Ambas devem ser strings.

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.

STRCAT

Sintaxe

```
strcat(string_destino, string_origem);
```

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

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.

STRLEN

Sintaxe:

```
int variável = strlen(string);
```

- Retorna o tamanho de uma string.

strcmp

Sintaxe:

```
int variável = strcmp(string1, string2);
```

Compara o conteúdo de duas strings. 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

STRSTR

Sintaxe:

```
char *strstr( const char* str, const char* substr );
```

- Retorna um ponteiro que indica a primeira ocorrência da string 'substr' na string 'str'.

MEMSET

Sintaxe:

```
void* memset( void* dest, int ch, size_t count );
```

- Coloca o caractere 'ch' nas 'count' primeiras posições da string 'dest'

MEMCPY

Sintaxe:

```
void* memcpy( void* dest, const void* src, size_t count );
```

- Copia os 'count' primeiros caracteres da string 'src' e coloca nas primeiras 'count' da string 'dest'.

STRREV

Sintaxe:

```
reverso = strrev(str1)
```

- Inverte a string s sobre ela mesma.