

**STRINGS****// Como C armazena uma string (string.c)**

#include &lt;stdio.h&gt;

```
void main(void)
{
    char string[256];
    int i;

    for (i = 0; i < 26; i++)
        string[i] = 'A' + i;
    string[i] = NULL;
    printf("A string contém %s\n", string);
}
```

**// Onde é finalizada uma string (finalizada.c)**

#include &lt;stdio.h&gt;

```
void main(void)
{
    char string[256];
    int i;

    for (i = 0; i < 26; i++)
        string[i] = 'A' + i;
    string[10] = NULL;           // finalizando a string
    printf("A string contém %s\n", string);
}
```

**// Usando strlen() (tamanho.c)**

#include &lt;stdio.h&gt;

#include &lt;string.h&gt;

```
void main(void)
{
    char titulo_livro[] = "Programação C/C++";
    printf("%s contém %d caracteres\n", titulo_livro, strlen(titulo_livro));
}
```

**// Copiando uma string para outra (copia.c)**

#include &lt;stdio.h&gt;

#include &lt;string.h&gt;

```
void main(void)
{
    char livro[128] = "Este texto será perdido com a cópia";
    strcpy(livro, "Programação C/C++");
    printf("Nome do livro: %s\n", livro);
}
```

**// Anexando o conteúdo de uma string em outra (anexa1.c)**

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

void main(void)
{
    char nome[64] = "Dálmata Macho";
    strcat(nome, " Happy");
    printf("O nome completo do Happy é %s\n", nome);
}
```

**// Anexando n caracteres em uma string (anexa2.c)**

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

void main(void)
{
    char nome[64] = "Fernando";
    strncat(nome, " e Rute", 8);
    printf("Você votou em %s?\n", nome);
}
```

**// Comparando duas strings (compara1.c)**

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

void main(void)
{
    printf("Comparando Abc com Abc %d\n", strcmp("Abc", "Abc"));
    printf("Comparando abc com Abc %d\n", strcmp("abc", "Abc"));
    printf("Comparando abcd com abc %d\n", strcmp("abcd", "abc"));
    printf("Comparando abcd com abce %d\n", strcmp("abcd", "abce"));
}
```

**// Comparando os primeiros n caracteres de duas strings (compara2.c)**

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

void main(void)
{
    printf("Comparando 3 letras Abc com Abc %d\n",
        strncmp("Abc", "Abc", 3));
    printf("Comparando 3 letras abc com Abc %d\n",
        strncmp("abc", "Abc", 3));
    printf("Comparando 3 letras abcd com abc %d\n",
        strncmp("abcd", "abc", 3));
    printf("Comparando 5 letras Abc com Abcd %d\n",
        strncmp("Abc", "Abcd", 5));
}
```

**// Convertendo a representação em string de caracteres de um número (converte.c)**

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

void main(void)
{
    int int_result;
    float float_result;
    long long_result;

    int_result = atoi("1234");
    float_result = atof("12345.678");
    long_result = atol("1234567L");

    printf("%d %f %ld\n", int_result, float_result, long_result);
}
```

**// Localizando uma substring dentro de uma string (localiza.c)**

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

void main(void)
{
    printf("Procurando Abc em AbcDef %s\n",
        (strstr("AbcDef", "Abc")) ? "Encontrado" : "Não encontrado");
    printf("Procurando Abc em abcDef %s\n",
        (strstr("abcDef", "Abc")) ? "Encontrado" : "Não encontrado");
    printf("Procurando Abc em AbcAbc %s\n",
        (strstr("AbcAbc", "Abc")) ? "Encontrado" : "Não encontrado");
}
```

**// Determinando o que é um caractere**

Função	O que realiza ?
isalnum(x)	determina se x é alfanumérico
isalpha(x)	determina se x é uma letra
isascii(x)	determina se x é um valor ASCII
isdigit(x)	determina se x é um dígito (0 a 9)
isgraph(x)	determina se x é um caracter gráfico
isupper(x)	determina se x é maiúsculo
islower(x)	determina se x é minúsculo

// Usando strtok (token.c)

```
#include <iostream>
```

```
using std::cout;
```

```
using std::endl;
```

```
#include <cstring>
```

```
int main()
```

```
{
```

```
    char string[] = "Esta frase possui 7 unidades léxicas (tokens)";
```

```
    char *tokenPtr;
```

```
    cout << "A string que vai ser dividida é:\n" << string
```

```
        << "\n\nOs tokens são:\n";
```

```
    tokenPtr = strtok( string, " " );
```

```
    while ( tokenPtr != NULL )
```

```
    {
```

```
        cout << tokenPtr << "\n";
```

```
        tokenPtr = strtok( NULL, " " );
```

```
    }
```

```
    return 0;
```

```
}
```