



## Funções de Manipulação de Strings:

Uma string é um *array* de caracteres terminado por um caracter nulo '\0' :

p	a	l	a	v	r	a	\0
---	---	---	---	---	---	---	----

Exemplo de declaração de strings:

```
char str1[30];  
char frase[150];  
char nome1[] = {'A','n','a',' ','M','a','r','i','a','\0'};  
char nome2[] = "Ana Maria";
```

As funções básicas para se manipular uma string estão prototipadas na biblioteca *string.h* e estão dispostas na tabela abaixo:

Funções Básicas	Descrição
strcpy(s1,s2)	Copia a string s2 na string s1.
strcmp(s1,s2)	Compara duas strings e determina a ordem (alfabética) das duas. Retorna 0 se ambas são iguais, maior que 0 se s1>s2, e menor que zero caso contrário.
strlen(s1)	Retorna o tamanho de uma string.
strcat(s1,s2)	Concatena s2 ao final de s1.

Os códigos a seguir ilustram o uso destas funções:

### Exemplo 1:

```
#include <string.h>  
#include <stdio.h>  
  
int main(){  
    int tam;  
    char str1[50], str2[50];  
    gets(str1);          /* leitura da string str1 */  
    tam = strlen("HELLO"); /* tam = 5 */  
    strcpy(str1,str2);   /* cópia de str1 em str2 */  
}
```

### Exemplo 2:

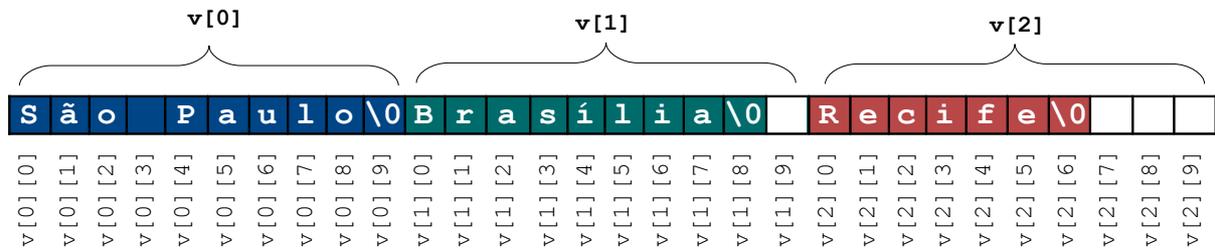
```
#include <string.h>  
#include <stdio.h>  
  
int main(){  
    char fonte[] = "Bom";  
    char dest[] = " dia!";  
    int tam1, tam2;  
    tam1 = strlen(fonte);    /* tam1 = 3 */  
    tam2 = strlen(dest);    /* tam2 = 5 */  
    strcat(fonte, dest);    /* dest = "Bom dia!" */  
}
```

Outras Funções	Descrição
<code>strncat(s1, s2, n)</code>	Concatena n caracteres da string2 na string1.
<code>strncmp(s1, s2, n)</code>	Compara os n primeiros caracteres de duas strings.
<code>strncpy(s1, s2, n)</code>	Copia os n primeiros caracteres da string2 na string1
<code>strcasecmp(s1, s2)</code>	Versão case insensitive de <code>strcmp()</code> .
<code>strncasecmp(s1, s2, n)</code>	versão case insensitive de <code>strncmp()</code> .

## Arrays de Strings

Arrays de strings são arrays de arrays de caracteres. Por exemplo, temos a declaração

```
char v[3][10];
```



O trecho de código abaixo pode ser utilizado para percorrer a matriz `v`, ilustrada acima:

```
for (i=0; i<3; i++)
    printf("%s\n", v[i]);
```

Resultado: São Paulo  
Brasília  
Recife

## Exercícios:

1) Analise os trechos de códigos a seguir e verifique se existe algum erro. Explique.

(a)

```
char a[], b[];
strcpy (a, "Corinthians");
strcpy (b, "Santos");
if (a < b)
    printf ("%s precede %s no dicionário", a, b);
```

(b)

```
char ola[] = "ola";
char ola2[] = "ola";
if(ola == ola2)
    printf("Iguais");
else
    printf("Nao sao iguais");
```

2) Seja o código abaixo que recebe uma *string* e devolve uma cópia da *string* com as letras minúsculas transformadas em maiúsculas:

```
void maius(char s[], char n[]);

main()
{
    char str[100], nstr[100];

    printf("Digite uma frase: ");
    scanf("%s",str); // Leitura de caracteres e espaço
    maius(str,nstr);
    printf("\nFrase em maiusculo: %s",nstr);
}

void maius(char s[], char n[]) {
    int i;
    for(i=0;s[i]!='\0';i++)
        n[i]=toupper(s[i]);
    n[i]='\0';
}
```

(a) Crie uma nova função que retorna o número de letras, o número de dígitos e o número de símbolos contido na string de entrada.

(b) Crie uma nova função que retorna apenas as letras transformadas para maiúsculo, descartando símbolos e dígitos.

(c) Crie uma nova função para substituir vários espaços em brancos por apenas um.

3) Faça um programa que lê uma string e retorna a string invertida.

4) Escreva um programa que lê uma string e retorna verdadeiro se a string de entrada é uma palíndrome.

5) Codifique um programa que recebe uma string e substitui cada letra pela letra seguinte circularmente (A é substituído por B, ... e Z é substituído por A). Escreva funções para codificar e decodificar strings.

6) Faça um programa que, a partir da digitação do infinitivo de um verbo regular, faça a conjugação do mesmo no presente do indicativo para as pessoas do singular e plural.

Exemplo: Entrada

```
CANTAR
Eu canto
Tu cantas
Ele canta
Nós cantamos
Vós cantais
Eles cantam
```

7) Escreva um programa que funciona como o jogo da forca. O usuário tenta acertar uma palavra (contida numa variável) usando 5 tentativas. A palavra é mostrada ao usuário com as letras marcadas com o símbolo '\*'. O usuário deve digitar uma letra (sem acentos, til, trema) e caso a letra esteja correta, o símbolo '\*' é substituído pela letra. Se o usuário não montar a palavra até a quinta tentativa, o jogo acaba em derrota.