

## VETOR DE CHAR (STRING)

Uma string (cadeia de caracteres) nada mais é do que um vetor do tipo char (vetor de caracteres), que armazena, normalmente uma mensagem ou frase. Além dessa particularidade, o último elemento da mensagem deve ser sempre um barra zero(\0). O último elemento da mensagem não precisa ser, necessariamente, o último elemento do vetor char.

### Exemplos:

```
char nome[11];  
char título[21] = {"Este é o título!"};
```

Acima pode-se destacar que a string "nome" poderá armazenar até 10 caracteres, pois um deve ser reservado para o terminador \0. A string "título" além de ser declarada, está sendo inicializada. Observe que o tamanho da mensagem é de 16 caracteres e a variável suporta até 20 (mais o terminador). Neste caso, o terminador é colocado após o caractere "!", posição [15], e não no final da string, índice [20].

### Leitura de string

a) Com a função scanf()

Utiliza a formatação %s (string).

#### Importante:

- a função scanf realiza a leitura até encontrar um espaço, colocando o caractere terminador \0;

Por exemplo: scanf("%s", palavra);

b) Com a função gets()

Utiliza a seguinte sintaxe:

```
gets (str);
```

Onde:

- str é a string que está sendo lida;

Com esta função é permitido ler frases.

### Exibição

puts() – função que exibe todo o conteúdo da string.

Exemplo: puts(str) → exibe o conteúdo da variável str.

### Funções da biblioteca string.h

- a função strcpy()

Sintaxe: strcpy(destino, origem);

copia a string-origem para a string- destino.

**Exemplo:**

```
strcpy (str2,str1); /* Cópia str1 em str2 */
```

```
strcpy (str3,"PROGII"); /* Cópia "PROGII" em str3 */
```

- a função **strncpy()** copia a string-origem para a string- destino, porém deve ser especificado o tamanho a ser copiado.

**Exemplo:**

```
strcpy(str1, "aula de progII online");  
strncpy(str2, str1, 12);
```

- a função **strcat()** concatena a segunda string ao final da primeira string.

**Exemplo:**

```
strcpy (str2,"C ");  
strcat (str2,str1); /* str2 armazenará C + o conteúdo de str1 */
```

- a função **strlen()** retorna o comprimento da string fornecida (sendo que o terminador não é contado). Exemplo:  
tamanho=strlen (str);

**Converter a string para maiúsculo ou minúsculo**

strupr( ) – converte a string para maiúsculo.

**Exemplo**

```
char nome[40]={“fipp – faculdade de informática”};
```

```
strupr(nome);  
resultará em: “FIPP – FACULDADE DE INFORMÁTICA”
```

strlwr() – converte a string para minúsculo.  
strlwr(nome); resultará em: “fipp – faculdade de informática”

**Pode-se usar também as funções toupper() e tolower()**

```
for (i=0; i<strlen(nome); i++)  
    nome[i]=toupper(nome[i]);
```

```
for (i=0; i<strlen(nome); i++)  
    nome[i]=tolower(nome[i]);
```

**Comparar duas strings**

- A função **strcmp()** compara a string 1 com a string 2.

Se as duas forem idênticas a função retorna zero.

**Exemplo:**

```
resultado=strcmp(str1,str2);
if (resultado == 0)
    printf("identicas);
else if (retorno < 0)
    printf("A string1 é menor que a string2.");
else
    printf("A string1 é maior que a string2.");
```

**strcmp( ) – Sintaxe: strcmp(str1, str2);**

Obs: Não faz diferença letras maiúscula e minúscula.

---

**Exercícios**

- 1) Ler uma frase e imprimir os 4 primeiros caracteres.
- 2) Ler uma frase, exibir a frase em letras maiúsculas, a quantidade de caracteres e depois apenas os caracteres que estão na posição ímpar do vetor.
- 3) Faça um programa que leia uma frase de até 60 caracteres. Imprimir a frase sem os caracteres em branco e também a quantidade de caracteres em branco.
- 4) Faça um programa que leia duas palavras, verifique o tamanho de cada palavra, concatenar as palavras e exibir.
- 5) Pesquisar as funções de string para converter string em float, string em int, float em string e int em string.