

String em C é um vetor contendo um conjunto de caracteres (letras, símbolos, números), em que o último caractere deve ser o '\0'.

Declaração:

```
char nome[] = "Unoeste";  
char nome[8];
```

Leitura:

scanf() – é limitada para a leitura de strings. Lê somente os caracteres até o espaço.

Exemplo: Leitura do nome "FIPP – Faculdade de Informática"

```
scanf("%s", nome);  
resultará em apenas: "FIPP"
```

#include<string.h>

gets() – função própria para leitura de string. Lê todos os caracteres até digitar ENTER.

```
gets(nome);  
resultará em: "FIPP – Faculdade de Informática"
```

Exibição:

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

```
puts(nome);  
resultará em: "FIPP – Faculdade de Informática"
```

```
puts(&nome[4]);  
resultará em: " – Faculdade de Informática"
```

printf() – a função printf exibe uma string usando a máscara de dados %s. Pode-se exibir uma mensagem juntamente com a string.

```
printf("Departamento: %s", nome);  
resultará em: "Departamento: FIPP – Faculdade de Informática"
```

Tamanho de uma string:

```
printf("Quantidade de caracteres: %d", strlen(nome));
```

Copiando uma string em outra:

strcpy() – Sintaxe: **strcpy(destino, origem);**

```
gets(nome);  
strcpy(nome2, nome);
```

Concatena duas strings:

strcat() – Sintaxe: **strcat(destino, origem);**

```
strcat(nome, sobrenome);
```

Converte a string para maiúsculo:

strupr() – converte a string para maiúsculo.

```
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()**

```
#include <ctype.h>
```

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

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

Compara duas strings e retorna um valor inteiro:

strcmp() – Sintaxe: **strcmp(string1, string2);**

Obs: Considera diferente letras maiúscula e minúscula.

```
gets(nome1);  
gets(nome2);  
retorno = strcmp(nome1, nome2);  
if (retorno == 0)  
    printf("As duas strings são iguais.");  
else  
if (retorno < 0)  
    printf("A string1 é menor que a string2.");  
else  
    printf("A string1 é maior que a string2.");
```

stricmp() – Sintaxe: **stricmp(string1, string2);**

Obs: Não faz diferença letras maiúscula e minúscula.

Converte string em números:

```
#include<stdlib.h>
char svalor[]="110";
int valor_i;
float valor_f;

valor_i = atoi(svalor); //converte de string para int
valor_f = atof(svalor); //converte de string para float
```

Converte números em strings:

```
char svalor_i[10];
char svalor_f[10];
int valor_i=10;
float valor_f=99.99;

itoa(valor_i, svalor_i, 10); //converte de int para string
sprintf(svalor_f, "%.2f", valor_f); //converte de float para string
```

Exercícios de Fixação:

1) Faça um algoritmo que leia uma frase e uma palavra, e conte quantas vezes essa palavra é encontrada na frase.

Exemplo:

frase: "Hoje de manha tem aula de ATP I"

palavra: "de"

Resultado:

A palavra "de" foi encontrada 2 vezes.