Lista de comandos para Strings

Gets

Lê da entrada padrão (stdin) até encontrar uma nova linha ou o fim de arquivo (EOF). A nova linha não é incluída na string lida. O caractere NULL ('\0') é automaticamente adicionado ao fim da string.

AVISO: essa função não é segura! Como não é possível especificar o número máximo de caracteres a serem lidos, é possível ler caracteres além do tamanho da string passada como parâmetro, causando uma falha de segurança conhecida como buffer overflow.

```
int main() {
   char nome[128];
   puts("Digite o seu nome: ");
   gets(nome); // inseguro. veja o aviso acima.
   printf("Seu nome é %s.\n", nome);
   return 0;
}
```

fflush()

fflush() é normalmente usado apenas para fluxo de saída. Sua finalidade é limpar (ou liberar) o buffer de saída e mover os dados do buffer para o console (no caso de stdout) ou disco (no caso de fluxo de saída de arquivo). Também pode ser utilizado o fflush(stdin);

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();

#include <string.h> //necessário para todos esses comandos

strcpy

Sintaxe:
Realiza a cópia do conteúdo de uma variável a outra.

```
int main (void)
{
char nome[15];
strcpy(nome, "Fulano de Tal");
//strcpy(string_destino, string_origem);
//note que a string de destino é nome
//a string de origem é "Fulano de Tal"
```

strncpy

Realiza a cópia do conteúdo de uma variável a outra, porém, deve ser especificado o tamanho a ser copiado.

```
int main (void)
{
char str1[8] = "Curso C";
char str2[5];
strncpy(str2, str1, 5);
```

strcat

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

```
int main (void)
{
char str[10] = "Curso";
strcat(str, " de C");
//Concatena a string " de C" com o conteúdo da string str
```

strncat

Realiza a concatenação do conteúdo de uma variável a outra, porém, deve ser especificado o tamanho a ser concatenado. Ambas devem ser strings.

```
int main (void)
{
char str1[20] = "Curso";
char str2[17] = " de programacao C";
strncat(str1, str2, 15);
//concatena a string1 com 15 posições da string2
printf("str1 = %s\n", str1);
//Será exibido Curso de Programação
```

strlen

Determina o tamanho de uma string.

```
int main (void)
{
char str[5] = "Curso";
int tamanho;
tamanho = strlen(str);
printf("O tamanho da string %s vale %d\n", str, tamanho);
```

strcmp

variável do tipo inteiro = strcmp(string1, string2);

Compara o conteúdo de duas strings;

Possíveis 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

```
int main (void)
{
char str1[4] = "abc";
char str2[4] = "abd";
int retorno;
retorno = strcmp(str1, str2);
printf("retorno = %d\n", retorno);
//mostra o retorno da função strcmp
```

strncmp

Também faz a comparação do conteúdo de duas strings, porém, deve ser especificado o tamanho a ser comparado;

Possíveis 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

```
int main (void)
{
char str1[15] = "Curso de C";
char str2[15] = "Curso de Java";
int retorno;
retorno = strncmp(str1, str2, 5);
```